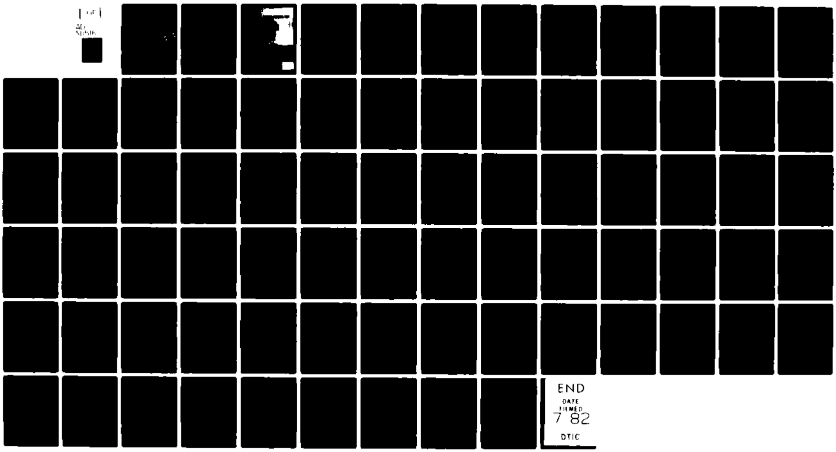
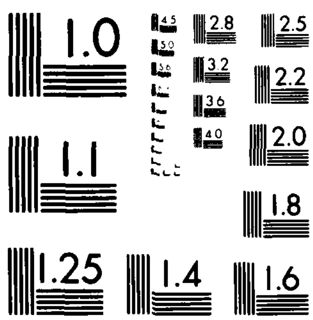
AD-A115 161 CALIFORNIA UNIV DAVIS SIGNAL AND IMAGE PROCESSING LAB F/G 12/1
A CLASS OF FFT BASED ALGORITHMS FOR LINEAR ESTIMATION.(U)
APR 82 A K JAIN, J JASIULEK DAAG29-78-G-0206
UNCLASSIFIED SIPL-81-5 ARO-16222.9-EL NL





MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 16222.9-EL	2. GOVT ACCESSION NO. AD-A115 161	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Class of FFT Based Algorithms for Linear Estimation		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Anil K. Jain Joachim Jasiulek		8. CONTRACT OR GRANT NUMBER(s) DAAG29 78 G 0206
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Davis, CA 95616		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE May 81; revised Apr 82
		13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

algorithms
estimating
fast fourier transform
matrices (mathematics)
filters
real time processes
image processing
signal processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In the past two decades since the advent of Kalman's recursive filter, numerous algorithms for linear estimation have emerged. Most of these algorithms are recursive and rely on solving a Riccati equation or equivalent recursive equations. It will be shown how some of the classical problems such as Linear Smoothing and Recursive Block Filtering problems can be solved exactly by some new nonrecursive algorithms which are based on the Fast Fourier Transform (FFT). Moreover, these algorithms are readily modified to generate the Riccati matrix at specified times, if this is desired. These results are then extended to a block filtering algorithm,

82 06 07 188

20. ABSTRACT CONTINUED

where data is received and smoothed recursively block by block. Real time batch processing applications include image processing and array processing of signals.

UCD

University of California, Davis

Signal & Image Processing
Laboratory

ABSTRACT

In the past two decades since the advent of Kalman's recursive filter, numerous algorithms for linear estimation have emerged. Most of these algorithms are recursive and rely on solving a Riccati equation or equivalent recursive equations. It will be shown how some of the classical problems such as Linear Smoothing and Recursive Block Filtering problems can be solved exactly by some new nonrecursive algorithms which are based on the Fast Fourier Transform (FFT). Moreover, these algorithms are readily modified to generate the Riccati matrix at specified times, if this is desired. These results are then extended to a block filtering algorithm, where data is received and smoothed recursively block by block. Real time batch processing applications include image processing and array processing of signals.

A Class of FFT Based Algorithms for
Linear Estimation

by

Anil K. Jain
and
Joachim Jasiulek

Report No. SIPL-81-5 May 1981

Research supported by the Army Research Office, Durham, North Carolina, under Grant DAAG29-78-G-0206.

Revised April 1982

1. INTRODUCTION

In the past two decades since the introduction of Kalman's recursive filter [1,2] several surveys of this subject [3-5] have exhibited different algorithms for the same basic filter. For example, Ho and Lee [6] use a Bayesian approach, Rauch et. al. [7] utilize a maximum likelihood principle, Meditch [8] uses a projection theorem, Kailath [9] employs innovations, and Lainiotis [10] uses the partition theorem to rederive the Kalman filter.

The conventional Kalman approach [1] involves the propagation of a state estimate and the error covariance matrix from stage to stage. Other approaches include the filtering algorithm by Fraser [11] which is based on finding the information matrix. A family of square root algorithms which recursively compute the square roots of the covariance matrix or of the information matrix is associated with Potter [12], Dyer and McReynolds [13], Schmidt [14], Kaminski and Bryson [15], and Bierman [16]. The Riccati equation plays a major part in all of these algorithms. Recently, Kailath [17] and Morf et al. [18], developed filtering and smoothing algorithms in which the Riccati equation is replaced by the computationally advantageous Chandrasekhar equation.

After Kalman proposed the filtering algorithm, the smoothing problem was subsequently solved in the state-space time domain by Carlton [19], Rauch [20], Bryson and Frazer [21], Rauch et. al. [17], Meditch [8], Mayne [22], Anderson et. al. [23], and many others.

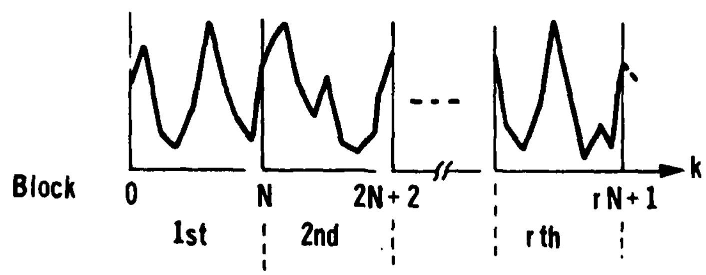
The numerous algorithms that have been derived are mostly recursive in structure and rely on solving the Riccati equation or equivalently the Levinson-Trench normal equations [24] associated with the solution of Toeplitz equations.

In this report we present some new algorithms for fixed interval smoothing, solution of Riccati equations, and block filtering problems that arise in linear estimation theory for discrete, time-invariant systems. These algorithms have several interesting features which make them attractive in the context of modern digital signal processing. These algorithms are non-recursive, fast and are based on the Fast Fourier Transform (FFT). For example, for ARMA models, typical recursive algorithms require $O(n^2N)$ operations, while the new algorithms which utilize the FFT need only $O((\log_2 N+n)N)$ operations to smooth N+1 samples of an n states, single input single output ARMA system with $N >> n$. Moreover, our algorithms seem to be less sensitive to round off and truncation errors. Finally since these new algorithms utilize the FFT and are non-recursive, they could be used to process large data batches efficiently in parallel, and would be well suited for VLSI architectures.

The smoothing algorithm developed here does not require solving the Riccati equation so that one is not confronted with the associated numerical problems such as insuring the positive semi-definiteness of the error covariance matrix e.g., as in the square root algorithms. It is shown that the optimal smooth estimate can be represented as a sum of two components. One component is the output of a Wiener filter with discrete frequency response. This Wiener filter is associated with a steady state periodic system which is observed for one period. The other component, called the boundary response, is determined completely by certain initial and terminal values of the observations and the Wiener filter output.

Even though the Riccati matrix is not required in our smoothing algorithms, it plays a fundamental role in a large number of problems. Many times it is desired to find the steady state solution of the Riccati equation associated with a linear, time-invariant system. In some situations, the given system is over-sampled with respect to the Nyquist rate, e.g., in radar systems, where the signal being observed may be a narrow band signal but the receiver bandwidth is much larger. Hence it may be desirable to obtain the Riccati matrix at the Nyquist rate, i.e., at equal lags of time. In section 5, it is shown that a minor modification of the smoothing algorithm yields the Riccati matrix at several instants of time via our FFT approach.

With the advent of array processors and array scanners, data is often received in blocks, batches, packets, arrays or lines. Hence it is desirable to consider filters which operate sequentially on blocks of data. Such filter structures have been considered in digital signal processing for convolution of a long sequence of data with a finite impulse response (FIR) filter. We introduce a new so called Recursive Block Filter. As the name suggests, this filter smoothes the data non-recursively within a block and recursively from block to block. Suppose the measurements are received in blocks of N+1 samples. One is asked to find the optimal smoothed estimate, $\hat{x}_0^i, \hat{x}_1^i, \ldots \hat{x}_N^i$ for $i = 1,2,\ldots$ given the observations, z_k for $k = 0,1, \ldots iN$ where i denotes the i^{th} block of the data shown below.



It is an on-line filter in the sense that if one treats a time-block as a unit or a packet of time, then one is asked to find the filter estimate $\hat{x}_{i|i}$ where i denotes the i^{th} block, or packet. It is also observed that this filter is similar to the fixed lag smoother, except that the lag of this filter is over non-overlapping samples.

This recursive block filter smoothes data block by block and can be implemented on-line, while preserving the nice properties of smoothing as well as all the foregoing advantages of the new algorithms. Also this recursive block filter has the same computational complexity as our new smoothing algorithm. In fact, by combining the techniques of the smoothing algorithm and that of calculating the Riccati matrix at equal lags, one easily realizes the recursive block filter.

Applications of the recursive block filter can be found in communications and telemetry where the fixed lag smoother is known to be useful, in image processing where a block of data is available at one time, and in digital on-line deconvolution of finite impulse response systems. Often in practice, a discrete time-varying system is modelled as a piecewise time invariant system. The recursive block filter can be extended to such models easily by simply applying our algorithms to successive time-invariant blocks.

Our results utilize the fact that the fixed interval smoother, Kalman filter and the associated Riccati equation can all be imbedded into a fundamental boundary value problem. In Section 2 we show the relationship of the various filters to their parent boundary value problem. In Section 3 we review the permuted controllable canonical form of state variable models which we use to develop our algorithms in Sections 4 and 5. Section 6 contains additional remarks. The smoothing algorithm for a special case is derived in Section 7. This provides a good introduction to the main ideas of the general derivation. This section also contains some numerical examples.

2. THE FIXED INTERVAL SMOOTHER

Consider the following discrete, time invariant system:

$$\underline{x}_{k+1} = A\underline{x}_k + B\varepsilon_k \tag{2.1}$$

$$z_k = C\underline{x}_k + \eta_k \tag{2.2}$$

where $\underline{x}_k \in R^n$, $\varepsilon_k \in R^m$, $z_k \in R^p$, and A, B, C are constant matrices of appropriate dimensions.

Assume that $\{\varepsilon_k\}$ and $\{\eta_k\}$ are independent, zero mean, Gaussian, white processes with covariances

$$E[\varepsilon_k \varepsilon_\ell^T] = K\delta_{k,\ell}$$

and

$$E[\eta_k \eta_\ell^T] = R\delta_{k,\ell}$$

where $\delta_{k,\ell} = 0$ if $k \neq \ell$ and $\delta_{k,k} = 1$. Also $\underline{x}_0$ is a zero mean Gaussian random variable of covariance Q_0 which is independent of $\{\varepsilon_k\}$ and $\{\eta_k\}$.

The fixed interval smoothing problem for this model consists of finding the best mean square estimate $\hat{\underline{x}}_k$ of $\underline{x}_k$ for $k = 0, \ldots N$, given all observations z_k, $k = 0, \ldots N$.

This problem is equivalent to maximizing the conditional probability

$$p(\hat{\underline{x}}_k | z_0, \ldots z_N) \text{ for all } k.$$

Applying Bayes rule and noting that all random variables are Gaussian, we get [e.g., 26,27] the problem of minimizing

$$J = \frac{1}{2}||\hat{\underline{x}}_0||^2_{Q_0^{-1}} + \frac{1}{2}\sum_{k=0}^{N} ||z_k - C\hat{\underline{x}}_k||^2_{R^{-1}} + \frac{1}{2}\sum_{k=0}^{N}||\varepsilon_k||^2_{K^{-1}}$$

subject to the state equations(2.1) and (2.2), where $||\cdot||_A$ denotes the norm induced by A. Using Lagrange multipliers this can be transformed into the following unconstrained problem

$$\min_{\varepsilon,\hat{x},\lambda} J + \sum_{k=0}^{N} \lambda_{k+1}^T [\hat{\underline{x}}_{k+1} - A\hat{\underline{x}}_k - B\varepsilon_k] .$$

The first order necessary conditions are obtained by differentiating with respect to $\hat{\underline{x}}_k$, ε_k and λ_{k+1}:

$$\lambda_k - C^T R^{-1}(z_k - C\hat{\underline{x}}_k) - A^T\lambda_{k+1} = 0 \tag{2.3}$$

$$K^{-1}\varepsilon_k - B^T\lambda_{k+1} = 0 \tag{2.4}$$

$$\hat{\underline{x}}_{k+1} = A\hat{\underline{x}}_k + B\varepsilon_k \tag{2.5}$$

for k = 1, ... N.

Substituting (2.4) into (2.5) results in

$$\hat{\underline{x}}_{k+1} = A\hat{\underline{x}}_k + BKB^T\lambda_{k+1} \text{ , } k = 0, \ldots , N \tag{2.6}$$

and rearranging (2.3) we find

$$\lambda_k = A^T\lambda_{k+1} + C^T R^{-1}(z_k - C\hat{\underline{x}}_k) \text{ , } k = 0, \ldots , N \tag{2.7}$$

where, in the boundary term for $k = 0$

$$Q_0^{-1}\hat{\underline{x}}_0 - C^T R^{-1}(z_0 - C\hat{\underline{x}}_0) - A^T\lambda_1 = 0,$$

we have defined $\lambda_0 \triangleq Q_0^{-1}\hat{\underline{x}}_0$. For $k = N$ we find $\lambda_{N+1} = 0$. Thus equations (2.6), (2.7) and

$$\hat{\underline{x}}_0 = Q_0\lambda_0 \text{ , } \lambda_{N+1} = 0 \tag{2.8}$$

define a two point boundary value problem which is equivalent to the original smoothing problem.

A standard approach to solving the smoothing problem is by constructing the smoothed estimate as a combination of a forward and backward filter [22].

Forward Filter:

(2.9) $$R_{k+1} = AR_kA^T + BKB^T - G_kH_kG_k^T;\ R_0 = Q_0$$

(2.10) $$s_{k+1} = As_k + G_kH_k[z_k - Cs_k];\ s_0 = 0$$

where

$$G_k = AR_kC^T \text{ and } H_k = [R + CR_kC^T]^{-1}\ .$$

Backward Filter:

(2.11) $$\lambda_k = [I - C^TH_kCR_k][A^T\lambda_{k+1} + C^TR^{-1}(z_k - Cs_k)]$$

$$\lambda_{N+1} = 0\ .$$

Then the smoothed estimate is given by

(2.12) $$\hat{\underline{x}}_k = R_k\lambda_k + s_k\ .$$

Remarks

i) This method is commonly referred to as the two sweep method.

ii) Here R_k is the solution of the matrix Riccati equation (2.9), and H_k is the covariance of the error $(z_k - Cs_k)$.

iii) The s_k obtained from (2.10) is the one step predictor $E[x_{k+1}|z_\ell, \ell \leq k]$ which arises in Kalman filtering.

3. REVIEW OF THE PERMUTED CONTROLLABLE CANONICAL FORM

The computations required in solving boundary value problems like (2.6) - (2.7) are simplified significantly when canonical forms are used. We will concentrate on the use of the permuted controllable form [28], but other canonical forms like the permuted observable form can also be used.

In the following brief review it will be assumed that the system

$$\underline{x}_{k+1} = A\underline{x}_k + Bu_k \tag{3.1}$$

$$z_k = Cx_k \tag{3.2}$$

is controllable, i.e., the controllability matrix

$$(B,\ AB,\ \ldots\ A^{n-1}B)$$

has maximal rank. The smallest positive integer $\gamma \le n$ such that the matrix

$$M_\gamma = (B,\ AB,\ \ldots\ A^{\gamma-1}B)$$

has maximal rank is called the controllability index. Then the permuted controllable canonical form can be expressed as

$$\begin{bmatrix} \underline{\bar{x}}^1 \\ \cdot \\ \cdot \\ \underline{\bar{x}}^\gamma \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & P_1 & & & 0 \\ \vdots & & P_2 & & \\ & & & \ddots & \\ 0 & & & & P_{\gamma-1} \\ A_1 & & \cdots & & A_\gamma \end{bmatrix} \begin{bmatrix} \underline{\bar{x}}^1 \\ \cdot \\ \cdot \\ \underline{\bar{x}}^\gamma \end{bmatrix}_k + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \underline{I}_m \end{bmatrix} \bar{u}_k \tag{3.3}$$

$$z_k = \bar{C}\,\underline{\bar{x}}_k \tag{3.4}$$

where we assume rank B = m without loss of generality. Here the matrices P_i are projection matrices of order $r_i x r_{i+1}$, and $r_i \le r_{i+1}$. This can be viewed as a vector ARMA model.

We will now describe the change of coordinates from (3.1) - (3.2) to

(3.3) - (3.4) defined by the transformations

$$\bar{A} = PAP^{-1}, \bar{B} = PBQ^{-1}, \bar{C} = CP^{-1}$$

and

$$\bar{x}_k = Px_k$$

$$\bar{u}_k = Qu_k \ .$$

The exposition follows [28]:

Step 1: Formation of the Controllability Matrix

(3.5) $$C_M = [B, AB, \ldots A^{n-1}B] \ .$$

Step 2: Choice of Linearly Independent Vectors

From C_M, a set of n independent vectors, forming the matrix U, are selected as follows:

(3.6) $$U = \left[b_1 \ b_2 \ldots b_m \mid Ab_1 \ Ab_2 \ldots Ab_m \mid \ldots \right]$$

where b_i is the i^{th} column of B.

Step 3: Formation of State Transformation Matrix P

From U^{-1}, a selection of rows, e_i for i = 1, ... m is then made, each row corresponding to the last vector in each group of b_i. Then the transformation matrix, P, is obtained by multiplying the set of the rows being selected by A, that is

$$P = \begin{bmatrix} e_1 \\ e_1 A \\ \vdots \\ e_1 A^{i_1} \\ e_2 \\ \vdots \\ e_2 A^{i_2} \\ \vdots \\ e_m A^{i_m} \end{bmatrix} \tag{3.7}$$

Step 4: Formation of $\overline{A}$, PB and $\overline{C}$

As discussed before, $\overline{A}$ and $\overline{C}$ can be calculated from

$$\overline{A} = P\, A\, P^{-1}$$

$$\overline{C} = C\, P^{-1}$$

while

$$PB = \left[\begin{array}{c|c|c|c} 0 & 0 & & 0 \\ \vdots & & \cdots & \vdots \\ 1 & x & & x \\ \hline 0 & 0 & & 0 \\ \vdots & & & \\ & 0 & & \\ x & 1 & & x \\ \hline \hline 0 & 0 & & 0 \\ & & & 0 \\ x & x & & 1 \end{array}\right]$$

Step 5: Formation of Input Transformation Matrix

The transformation of input requires finding the matrix

$$Q = \begin{bmatrix} 1 & x & x & \cdots & x \\ x & 1 & x & \cdots & x \\ \cdot & & & \cdots & \cdot \\ \cdots & & & & \cdot \\ \cdot & & & & \cdot \\ x & x & x & \cdots & 1 \end{bmatrix}$$

where each successive row of Q corresponds to the non-zero rows of the matrix PB.

Step 6: Formation of $\overline{B}$

Having found the inverse of Q, $\overline{B}$ can be computed from

$$\overline{B} = P\,B\,Q^{-1}$$

Step 7: Permutation of States

The final form of the system is obtained by permutation.

Example:

To illustrate the procedure, an example is given below:

Assume the original system has the A,B,C, matrices as follows:

$$A = \begin{bmatrix} 0 & 0 & -4 & 0 & 4 \\ 1 & 0 & -8 & 0 & -4 \\ 0 & 1 & -5 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} . \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 & -4 & 1 & 0 \\ 1 & -1 & 1 & 0 & 1 \end{bmatrix}$$

The controllability matrix, U is found as

$$U = \begin{bmatrix} b_1 & b_2 & Ab_1 & Ab_2 & A^2b_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$U^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Because A^2b_1 and $A\ b_2$ are the fifth and fourth columns respectively, e_1 and e_2 are chosen to be the fifth and fourth rows of U^{-1} respectively.

The transformation matrix, P, is computed next as:

$$P = \begin{bmatrix} e_1 \\ e_1A \\ e_1A^2 \\ e_2 \\ e_2A \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -5 & 0 & -1 \\ 1 & -5 & 17 & -1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

and

$$P^{-1} = \begin{bmatrix} 8 & 5 & 1 & 4 & 1 \\ 5 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

After the transformation, the system has the $\bar{A}$, $\bar{C}$ matrix as:

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -4 & -8 & -5 & -4 & -3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\overline{C} = \begin{bmatrix} 1 & 1 & 0 & 2 & 1 \\ -4 & -1 & 0 & 0 & 0 \end{bmatrix}$$

while

$$PB = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

A further transformation of the input vector will yield the desired system matrix $\overline{B}$ as:

$$\overline{B} = PBQ^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

where

$$Q^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The final step is done as follows: Let the states of the transformed system be labelled as:

$$\overline{x}_k = [\overline{x}_k^1(1)\ \overline{x}_k^1(2)\ \overline{x}_k^1(3) \ \vdots\ \overline{x}_k^2(1)\ \overline{x}_k^2(2)]^T$$

Therefore by permutation three subgroups will be formed. They are:

$$x_k = \left[\overline{x}_k^1(1) \ \vdots\ \overline{x}_k^1(2)\ \overline{x}_k^2(1) \ \vdots\ \overline{x}_k^1(3)\ \overline{x}_k^2(2)\right]^T$$

The system will have the form of

$$\begin{bmatrix} \tilde{x}^1_{k+1} \\ \tilde{x}^2_{k+1} \\ \tilde{x}^3_{k+1} \end{bmatrix} = \left[\begin{array}{c|cc|cc} 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline -4 & -8 & -4 & -5 & -3 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} \tilde{x}^1_k \\ \tilde{x}^2_k \\ \tilde{x}^3_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \bar{u}_k$$

$$z_k = \begin{bmatrix} 1 & 1 & 2 & 0 & 1 \\ -4 & -1 & 0 & 0 & 0 \end{bmatrix} \tilde{x}_k$$

Therefore in this example

$$P_1 \triangleq [1 \quad 0] \;,\; P_2 \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\bar{A}_1 \triangleq \begin{bmatrix} -4 \\ 0 \end{bmatrix} ,\; \bar{A}_2 \triangleq \begin{bmatrix} -8 & -4 \\ 0 & 0 \end{bmatrix} ,\; \bar{A}_3 \triangleq \begin{bmatrix} -5 & -3 \\ 0 & -1 \end{bmatrix}$$

$$\bar{C}_1 \triangleq \begin{bmatrix} 1 \\ -4 \end{bmatrix} ,\; \bar{C}_2 \triangleq \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix} ,\; \bar{C}_3 \triangleq \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

For the case of a single input, single output system, the transformed $\bar{A}$, $\bar{B}$ matrices are more simple, and will take the following forms:

$$\bar{A} = \begin{bmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & 1 \\ a_1 & a_2 & \cdots & a_n \end{bmatrix} ,\; \bar{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

The method to find the transformation matrix, P, in this case is slightly different and is given as follows:

Let the characteristic polynomial of the matrix A be

$$\Delta(\lambda) = \mathrm{Det}(\lambda I - A) = \lambda^n + \alpha_1 \lambda^{n-1} + \ldots + \alpha_{n-1}\lambda + \alpha_n \,.$$

Then from the controllability matrix $[B \; AB \; \ldots \; A^{n-1}B]$ and the coefficients of the characteristic polynomial, the inverse of the transformation matrix, P^{-1}, is found as

$$P^{-1} = [q_1 \ q_2 \ \dots \ q_n]$$

where

$$\begin{aligned}
q_n &= B \\
q_{n-1} &= Aq_n + \alpha_1 q_n = AB + \alpha_1 B \\
q_{n-2} &= Aq_{n-1} + \alpha_2 q_n = A^2B + \alpha_1 AB + \alpha_2 B \\
&\vdots \\
q_1 &= Aq_2 + \alpha_n q_n = A^{n-1}B + \alpha_1 A^{n-2}B + \dots + \alpha_{n-1}B
\end{aligned}$$

Notice that there is no need to perform any permutation on the states since the transformed system is already in the desired form.

4. THE SOLUTION OF A TWO POINT BOUNDARY VALUE PROBLEM VIA THE FFT

Now we turn to the method of solving the boundary value problem associated with the fixed interval smoother. The following two point boundary value problem is considered.

$$(4.1)\quad \begin{bmatrix} \underline{x}^1 \\ \vdots \\ \underline{x}^\gamma \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & P_1 & & 0 \\ \vdots & & \ddots & \\ 0 & & 0 & P_{\gamma-1} \\ A_1 & \dots & & A_\gamma \end{bmatrix} \begin{bmatrix} \underline{x}^1 \\ \vdots \\ \underline{x}^\gamma \end{bmatrix}_k + \begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & \ddots & & \vdots \\ & & 0 & \\ 0 & \dots & 0 & K \end{bmatrix} \begin{bmatrix} \underline{\lambda}^1 \\ \vdots \\ \underline{\lambda}^\gamma \end{bmatrix}_{k+1}$$

$$(4.2)\quad \begin{bmatrix} \underline{\lambda}^1 \\ \vdots \\ \underline{\lambda}^\gamma \end{bmatrix}_k = \begin{bmatrix} 0 & & & A_1^T \\ P_1^T & & 0 & \vdots \\ & \ddots & & \\ 0 & & P_{\gamma-1}^T & A_\gamma^T \end{bmatrix} \begin{bmatrix} \underline{\lambda}^1 \\ \vdots \\ \underline{\lambda}^\gamma \end{bmatrix}_{k+1} + C^T R^{-1}(z_k - C\underline{x}_k) \qquad k = 0, \dots N$$

with boundary conditions

$$\underline{x}_0 = Q_0 \underline{\lambda}_0 + \mu_0$$

and

$$\underline{\lambda}_{N+1} = w$$

where Q_0, μ_0 and w are assumed to be specified. The problem (2.6) - (2.8) in controllable canonical form is a special case with the choices $\mu_0 = 0$ and $\underline{\lambda}_{N+1} = 0$. These general boundary values are needed when the Riccati matrix and block filter algorithms are derived. The P_i are projection matrices and $C = (C_1, \dots C_\gamma)$ will be partitioned accordingly.

For convenience we define I_m to be the identity matrix of order m, $P_0 \triangleq 0$, $P_\gamma \triangleq I_m$, $x_k \triangleq \underline{x}_k^\gamma$, $\lambda_k \triangleq \underline{\lambda}_k^\gamma$, $\lambda_k^0 \triangleq 0$ for all k, $\bar{A}_{\gamma+1} \triangleq -I_m$, $C_{\gamma+1} \triangleq 0$, and $C_0 \triangleq 0$.

We will now discuss how the problem (4.1) - (4.2) can be solved in terms of the m dimensional vectors x_ℓ rather than the n dimensional state vectors $\underline{x}_\ell$. It follows from (4.1) that

$$(4.3)\qquad \underline{x}_{k+1}^i = P_i \underline{x}_k^{i+1} \quad \text{for } i < \gamma ,$$

Applying (4.3) recursively results in

(4.4) $$\underline{x}_k^i = \tilde{P}_i x_{k-\gamma+i} \quad \text{for } i < \gamma \text{ , } k \geq \gamma - i$$

where

$$\tilde{P}_i \triangleq \prod_{j=i}^{\gamma-1} P_j \text{ .}$$

For later convenience, this expression will be extended to all $k \geq 0$. This can be done by using (4.4) as a definition for x_ℓ, when $\ell < 0$. The result is an equivalence class of vectors for which

$$x_0^i = \tilde{P}_i x_{-\gamma+i}, \; i < \gamma \text{ .}$$

Note that this is only a convenient notation, and does not mean that the original model is extended backwards in time.

From (4.1) we get the last component vector $\underline{x}_{k+1}^\gamma \triangleq x_{k+1}$ as

(4.5) $$x_{k+1} = \sum_{i=1}^{\gamma} A_i \underline{x}_k^i + K\lambda_{k+1} \text{ .}$$

Using (4.4), $\overline{A}_{\gamma+1} \triangleq -I_m$, and $\overline{A}_i \triangleq A_i \tilde{P}_i$, where $\tilde{P}_\gamma = P_\gamma = I_m$, this becomes

(4.6) $$x_{k+1} = \sum_{i=1}^{\gamma} \overline{A}_i x_{k-\gamma+i} + K\lambda_{k+1}$$

or after solving for λ_{k+1}:

(4.7) $$\lambda_{k+1} = -K^{-1} \sum_{i=1}^{\gamma+1} \overline{A}_i x_{k-\gamma+i}, \; 0 \leq k \leq N-1.$$

Similarly, equation (4.2) can be expressed componentwise as

(4.8) $$\underline{\lambda}_k^i = A_i^T \lambda_{k+1} + P_{i-1}^T \underline{\lambda}_{k+1}^{i-1} + f_k^i \text{ , } i = 1, \ldots \gamma$$

where $\underline{\lambda}_k^0 \triangleq 0$ by definition and $f_k \triangleq C^T R^{-1}(z_k - C\underline{x}_k)$ is partitioned as

(4.9) $$f_k^i = C_i^T R^{-1}(z_k - C\underline{x}_k)$$

or in terms of x_k as

(4.10) $$f_k^i = C_i^T R^{-1}(z_k - \sum_{j=1}^{\gamma} \bar{C}_j x_{k-\gamma+j}) \ ,$$

where $\bar{C}_j \hat{=} C_j \tilde{P}_j$. Applying (4.8) recursively we obtain (see (A3))

(4.11) $$\lambda_k = \sum_{i=0}^{\gamma-1} (\bar{A}_{\gamma-i}^T \lambda_{k+i+1} + \tilde{P}_{\gamma-i}^T f_{k+i}^{\gamma-i}) \ .$$

Substitution of (4.7) and (4.10) into (4.11) results in

(4.12) $$\sum_{i=-1}^{\gamma-1} \sum_{j=1}^{\gamma+1} D_{\gamma-i,j} x_{k-\gamma+j+i} = \tilde{z}_k^o \ , \ 1 \le k \le N-\gamma$$

where

(4.13) $$D_{i,j} \hat{=} \bar{A}_i^T K^{-1} \bar{A}_j + \bar{C}_i^T R^{-1} \bar{C}_j \ ; \ \bar{C}_{\gamma+1} \hat{=} 0$$

and

(4.14) $$\tilde{z}_k^o = \sum_{i=0}^{\gamma-1} \bar{C}_{\gamma-i}^T R^{-1} z_{k+i} \ .$$

After an index transformation (see (A1)), it is possible to collect terms involving $x_{k+\ell}$ as

(4.15) $$\sum_{\ell=-\gamma}^{\gamma} \tilde{A}_\ell x_{k+\ell} = \tilde{z}_k^o \ , \ 1 \le k \le N-\gamma \ .$$

Here

(4.16) $$\tilde{A}_\ell \hat{=} \sum_{j=\ell+1}^{\gamma+1} D_{j-\ell,j} \quad \text{for } \ell \ge 0$$

and

(4.17) $$\tilde{A}_\ell = \tilde{A}_{-\ell}^T \quad \text{for } \ell \le 0 \ .$$

If we write out (4.15) for $k = 1, \ldots N-\gamma$ we obtain the system

(4.18)
$$\begin{bmatrix} \tilde{A}_{-\gamma} \cdots \tilde{A}_{\gamma} & & 0 \\ & \ddots & \ddots \\ 0 & & \tilde{A}_{-\gamma} \cdots \tilde{A}_{\gamma} \end{bmatrix} \begin{bmatrix} x_{-\gamma+1} \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{z}^o_1 \\ \vdots \\ \tilde{z}^o_{N-\gamma} \end{bmatrix}$$

By putting the first and last γ subvectors on the right hand side, we obtain a perturbed block Toeplitz system:

(4.19)
$$\begin{bmatrix} \tilde{A}_0 \cdots \tilde{A}_{\gamma} & & 0 \\ \vdots & \ddots & \tilde{A}_{\gamma} \\ \tilde{A}_{-\gamma} & \ddots & \\ 0 & \tilde{A}_{-\gamma} & \tilde{A}_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{N-\gamma} \end{bmatrix} = \begin{bmatrix} \tilde{z}^o_1 \\ \vdots \\ \tilde{z}^o_{N-\gamma} \end{bmatrix} + \begin{bmatrix} E\tilde{x}^i \\ 0 \\ \vdots \\ 0 \\ E^T\tilde{x}^t \end{bmatrix}$$

where
$$E \hat{=} - \begin{bmatrix} \tilde{A}_{-\gamma} \cdots \tilde{A}_{-1} \\ & \ddots & \vdots \\ 0 & & \tilde{A}_{-\gamma} \end{bmatrix}, \quad \tilde{x}^i \hat{=} \begin{bmatrix} x_{-\gamma+1} \\ \vdots \\ x_0 \end{bmatrix}, \quad \tilde{x}^t \hat{=} \begin{bmatrix} x_{N-\gamma+1} \\ \vdots \\ x_N \end{bmatrix}$$

If we also define

$$\hat{x}^i \hat{=} \begin{bmatrix} x_1 \\ \vdots \\ x_{\gamma} \end{bmatrix} \text{ and } \hat{x}^t \hat{=} \begin{bmatrix} x_{N-2\gamma+1} \\ \vdots \\ x_{N-\gamma} \end{bmatrix}$$

then the boundary conditions lead to equations of the form (see appendix)

(4.20)
$$G^0\tilde{x}^i = G^1\hat{x}^i + \hat{z}^i$$

and

(4.21)
$$T^0\tilde{x}^t = T^1\hat{x}^t + \hat{z}^t .$$

where $\hat{z}^i$ and $\hat{z}^t$ depend on the observations and are defined in the appendix (also see Section 5). These equations can be used to convert (4.19) into a system involving $x_1, \ldots x_{N-\gamma}$ only:

$$(4.22)\quad \begin{bmatrix} \tilde A_0 & \cdots & \tilde A_\gamma & & 0 \\ \vdots & \ddots & & \ddots & \\ \tilde A_{-\gamma} & & \ddots & & \tilde A_\gamma \\ & \ddots & & \ddots & \vdots \\ 0 & & \tilde A_{-\gamma} & \cdots & \tilde A_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{N-\gamma} \end{bmatrix} = \begin{bmatrix} \tilde z_1 \\ \vdots \\ \tilde z_{N-\gamma} \end{bmatrix} + \begin{bmatrix} E(G^0)^{-1}(G^1\hat x^i + \hat z^i) \\ 0 \\ \vdots \\ 0 \\ E^T(T^0)^{-1}(T^1\hat x^t + \hat z^t) \end{bmatrix}$$

In order to solve this perturbed block Toeplitz system via the FFT, it will be rewritten as the perturbed block circulant system

$$(4.23)\qquad H\underline{\tilde x} = \underline{\tilde z} + J\Psi \begin{bmatrix} \hat x^i \\ \hat x^t \end{bmatrix}$$

where

$$H \hat= \begin{bmatrix} \tilde A_0 & \tilde A_1 & \cdots & \tilde A_\gamma & 0 & & 0 & \tilde A_{-\gamma} & \cdots & \tilde A_{-1} \\ \tilde A_{-1} & \tilde A_0 & & & \ddots & & 0 & & \ddots & \vdots \\ \vdots & & \ddots & & & \ddots & & & & \tilde A_{-\gamma} \\ \tilde A_{-\gamma} & & & \ddots & & & \ddots & & & \\ & \ddots & & & \ddots & & & \ddots & & \tilde A_\gamma \\ \tilde A_\gamma & & 0 & & & \ddots & & & \ddots & \vdots \\ \vdots & \ddots & & & & & \ddots & & & \tilde A_1 \\ \tilde A_1 & \cdots & \tilde A_\gamma & & & & \tilde A_{-\gamma} & \cdots & & \tilde A_0 \end{bmatrix}$$

$$\underline{\tilde x} \hat= \begin{bmatrix} x_1 \\ \vdots \\ x_{N-\gamma} \end{bmatrix}, \quad \underline{\tilde z} \hat= \begin{bmatrix} \tilde z_1^0 \\ \vdots \\ \tilde z_{N-\gamma}^0 \end{bmatrix} + \begin{bmatrix} E(G^0)^{-1}\hat z^i \\ 0 \\ \vdots \\ 0 \\ E^T(T^0)^{-1}\hat z^t \end{bmatrix}$$

$$J \hat= \begin{bmatrix} I_{m\gamma} & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & I_{m\gamma} \end{bmatrix} \text{ and } \Psi \hat= \left[\begin{array}{c|c} E(G^0)^{-1}G^1 & -E \\ \hline -E^T & E^T(T^0)^{-1}T^1 \end{array}\right]$$

Eqn. (4.23) is solved in two stages. First the $2m\gamma$ boundary terms

$$b = \begin{bmatrix} \hat x^i \\ \hat x^t \end{bmatrix}$$

are determined, and then $\tilde{\underline{x}}$ can be obtained by circulant deconvolution [28]. When (4.23) is multiplied by $J^T H^{-1}$ we get

$$b = J^T H^{-1} \tilde{\underline{z}} + J^T H^{-1} J \Psi b \quad \text{or} \tag{4.24}$$

$$b = [I_{2m\gamma} - J^T H^{-1} J \Psi]^{-1} J^T H^{-1} \tilde{\underline{z}} \ . \tag{4.25}$$

Finally, $\underline{x}$ is obtained from (4.23) and (4.25) as

$$\underline{x} = H^{-1} \tilde{\underline{z}} + H^{-1} J \Psi b \ . \tag{4.26}$$

Eqn. (4.25) requires inversion of only a $2m\gamma \times 2m\gamma$ matrix; $H^{-1}\tilde{\underline{z}}$ is a circular deconvolution that can be determined via the FFT. Once b is known, then (4.26) gives $\underline{x}$ easily. Since $J\Psi b$ is a sparse vector containing only $2m\gamma$ non-zero entries, $H^{-1}J\Psi b$ can be computed either directly if $2m\gamma \ll \log_2 Nm$ or via another FFT based circular deconvolution.

Remarks

There is an alternative to the approach presented which transforms (4.15) into a perturbed block Toeplitz system by extending the coefficient matrix to a square matrix:

$$\begin{bmatrix} \tilde{A}_0 & \cdots & \tilde{A}_\gamma & & 0 \\ \vdots & \ddots & & \ddots & \\ \tilde{A}_{-\gamma} & & \ddots & & \tilde{A}_\gamma \\ & \ddots & & \ddots & \vdots \\ 0 & & \tilde{A}_{-\gamma} & \cdots & \tilde{A}_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{z}_1^o \\ \cdot \\ \cdot \\ \tilde{z}_{N-\gamma}^o \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} E\tilde{x}^i \\ 0 \\ \vdots \\ 0 \\ \begin{pmatrix} \tilde{A}_{-\gamma} & \cdots & \tilde{A}_{\gamma-1} \\ & \ddots & \vdots \\ 0 & \tilde{A}_{-\gamma} & \tilde{A}_0 \end{pmatrix} \begin{pmatrix} x_{N-2\gamma+1} \\ \vdots \\ x_N \end{pmatrix} \end{bmatrix}$$

The dimensionality of the system increases, but if N is a highly composite number there may be advantages in speed when the FFT is computed.

5. ALGORITHMS

Initialization

The proposed algorithm to solve the two point boundary value problem and other related problems will now be stated. We start by summarizing the definitions of the necessary quantities.

$$x_k \hat{=} \underline{x}_k^\gamma \ , \ \lambda_k \hat{=} \underline{\lambda}_k^\gamma \ , \ P_0 \hat{=} 0 \ , \ P_\gamma \hat{=} I_m ,$$

$$\bar{A}_{\gamma+1} \hat{=} -I_m \ , \ C_{\gamma+1} \hat{=} 0$$

$$\tilde{x}^i \hat{=} \begin{bmatrix} x_{-\gamma+1} \\ \vdots \\ x_0 \end{bmatrix} , \ \hat{x}^i \hat{=} \begin{bmatrix} x_1 \\ \vdots \\ x_\gamma \end{bmatrix} , \ \tilde{x}^t \hat{=} \begin{bmatrix} x_{N-\gamma+1} \\ \vdots \\ x_N \end{bmatrix} , \ \hat{x}^t \hat{=} \begin{bmatrix} x_{N-2\gamma+1} \\ \vdots \\ x_{N-\gamma} \end{bmatrix}$$

$$\tilde{P}_i \hat{=} \prod_{j=i}^{\gamma-1} P_j \ , \ \bar{A}_i \hat{=} A_i \tilde{P}_i \ , \ \bar{C}_i \hat{=} C_i \tilde{P}_i$$

Note that $\tilde{P}_i$ contains only zeros and ones, and thus products with $\tilde{P}_i$ as a factor are not implemented as matrix multiplications.

$$D_{i,j} \hat{=} \bar{A}_i^T K^{-1} \bar{A}_j + \bar{C}_i^T R^{-1} \bar{C}_j$$

$$\Gamma_{0,\ell} \hat{=} \Gamma_{i,0} = 0 \ ; \ \Gamma_{i+1,\ell+1} \hat{=} \Gamma_{i,\ell} + D_{i+1,\ell+1} \ , \ 0 \le i,\ell \le \gamma-1$$

$$\tilde{A}_\ell \hat{=} \Gamma_{\gamma-\ell,\gamma} - \bar{A}_{\gamma-\ell+1}^T K^{-1} \ , \ 0 \le \ell \le \gamma$$

$$\tilde{A}_\ell \hat{=} \tilde{A}_{-\ell}^T \quad \text{for} \quad -\gamma \le \ell \le 0$$

H = block circulant matrix of order $N-\gamma$ whose first row is

$(\tilde{A}_0, \tilde{A}_1, \ldots \tilde{A}_\gamma, 0, \ldots 0, \tilde{A}_{-\gamma}, \ldots \tilde{A}_{-1})$

$$S \hat{=} \mathrm{diag}(\tilde{P}_1^T, \ldots \tilde{P}_\gamma^T)Q_0^{-1}\,\mathrm{diag}(\tilde{P}_1 \ldots, \tilde{P}_\gamma)$$

and let
$$\hat{S} = [\mathrm{diag}(\tilde{P}_1^T, \ldots \tilde{P}_\gamma^T)Q_0^{-1}]$$

$$G^0_{i,\ell} \hat{=} \Gamma_{i,\ell} + S_{i,\ell}$$

$$E_{i,\ell} \hat{=} \begin{cases} -\tilde{A}_{-\gamma-i+\ell} & \text{if } i \le \ell \\ 0 & \text{if } i > \ell \end{cases}$$

$$G^1 \hat{=} E^T$$

$V_0 \hat{=} (0 \ldots K)$, $V_{i+1} \hat{=} [V_i A^T]$, where the V_i are mxn matrices.

$$W_i \hat{=} V_i C^T R^{-1} \;,\; i = 0, \ldots \gamma-1$$

$$F_{0,j} \hat{=} F_{i,\gamma+1} \hat{=} 0 \;;\; F_{i+1,j} \hat{=} F_{i,j+1} + W_i\bar{C}_j \quad \text{for } 0 \le i < \gamma \,,\; 1 \le j \le \gamma$$

$$T^0_{i,j} \hat{=} \begin{cases} -\bar{A}_{i+j} + F_{i,j} & \text{if } 1 \le j \le \gamma-i+1 \\ F_{i,j} & \text{otherwise} \end{cases}$$

$$T^1_{i,j} = \begin{cases} 0 & j < \gamma-i+1 \\ \bar{A}_1 & j = \gamma-i+1 \\ \bar{A}_{i-\gamma+j} - \sum_{\ell=1}^{j-\gamma+i-1} W_{i-1+j-\gamma-\ell}\bar{C}_\ell & \text{otherwise} \end{cases}$$

$$\Psi_{11} = E(G^0)^{-1}G^1 \;,\; \Psi_{12} = -E$$

$$\Psi_{21} = -E^T \;,\; \Psi_{22} = E^T(T^0)^{-1}T^1$$

Find the inverse of the block circulant matrix H, and then perform the LU-decomposition of

$$\Phi = I_{2m\gamma} - J^T H^{-1} J\Psi$$

(i) Algorithm for the Boundary Value Problem (Figure 1)

Step 1: Perform the FIR filter operation on the observations z_k to obtain the sequence

$$\tilde{z}_k^0 = \sum_{j=0}^{\gamma-1} \bar{C}_{\gamma-j}^T R^{-1} z_{k+j} \ , \ 1 \le k \le N-\gamma$$

Step 2: Extract the initial and terminal variables $\hat{z}^i$ and $\hat{z}^t$:

$$\hat{z}_k^i = \sum_{j=0}^{k-1} \bar{C}_{k-j}^T R^{-1} z_j + [\hat{S}\mu_0]_k \ ; \ 1 \le k \le \gamma$$

$$\hat{z}_k^t = \sum_{j=0}^{k-1} W_{k-j-1} z_{N-j} + V_k \underline{\lambda}_{N+1} \ ; \ 1 \le k \le \gamma$$

Step 3: Compute

$$\underline{\tilde{z}}^b = J \begin{bmatrix} E(G^0)^{-1}\hat{z}^i \\ E^T(T^0)^{-1}\hat{z}^t \end{bmatrix}, \ \underline{\tilde{z}} = \underline{\tilde{z}}^b + \underline{\tilde{z}}^o$$

Step 4: Perform the block circular deconvolution

$$\underline{y} = H^{-1}\underline{\tilde{z}}$$

Step 5: Find

$$b = \Phi^{-1} J^T \underline{y}$$

Step 6: Determine the estimate $\underline{x}$ as

$$\underline{x} = \underline{y} + H^{-1} J\Psi b$$

by circular deconvolution.

Step 7: Determine the boundary value estimates

$$\tilde{x}^i = (G^0)^{-1}[G^1\hat{x}^i + \hat{z}^i]$$

$$\tilde{x}^t = (T^0)^{-1}[T^1\hat{x}^t + \hat{z}^t]$$

(ii) The Fixed Interval Smoothing Algorithm

If we let $\mu_0 = \underline{\lambda}_{N+1} = 0$ in Fig. 1, we arrive at the fixed interval smoothing algorithm. The observations are passed through an FIR filter with impulse response $\hat{C}_{\gamma+k} = \bar{C}^T_{\gamma+k}R^{-1}$, $-\gamma \le k \le 0$. The resulting signal $\{z^0_k\}$ plus $\{z^b_k\}$ is deconvolved by H^{-1} which is a circular convolution with the elements of the first column of the inverse of the block circulant matrix H (see Appendix B). Boundary elements of this filter output are extracted by the projection J^T, multiplied by boundary filter gain $\Psi\Phi^{-1}$, and then injected into a larger vector, which is again deconvolved by H^{-1} to obtain the boundary response y^b_k. The final estimate x_k is obtained by summing the responses y_k and y^b_k.

(iii) Algorithm for Solving the Riccati Equation via the FFT

The Riccati matrix R_k (see (2.9)) can also be found from (4.1), (4.2), if it is only desired to compute R_{N+1} for fixed N. To derive this method, suppose $z_k = 0$ for all k. Then (2.9) implies that $s_k = 0$ for all k. If we set $\lambda_i = e_i$, the ith unit vector, then (2.12) implies that

$$\hat{x}_{N+1} = R_{N+1}e_i ,$$

i.e., the ith column of R_{N+1}, and hence the matrix R_{N+1} can be obtained as solution of the following matrix version of (4.1), (4.2):

(5.1) $$\underline{X}_{k+1} = A\underline{X}_k + BKB^T\underline{\Lambda}_{k+1} \; ; \; \underline{X}_0 = Q_0\underline{\Lambda}_0$$

(5.2) $$\underline{\Lambda}_k = A^T\underline{\Lambda}_{k+1} - C^TR^{-1}C\underline{X}_k \; , \; \underline{\Lambda}_{N+1} = I_n$$

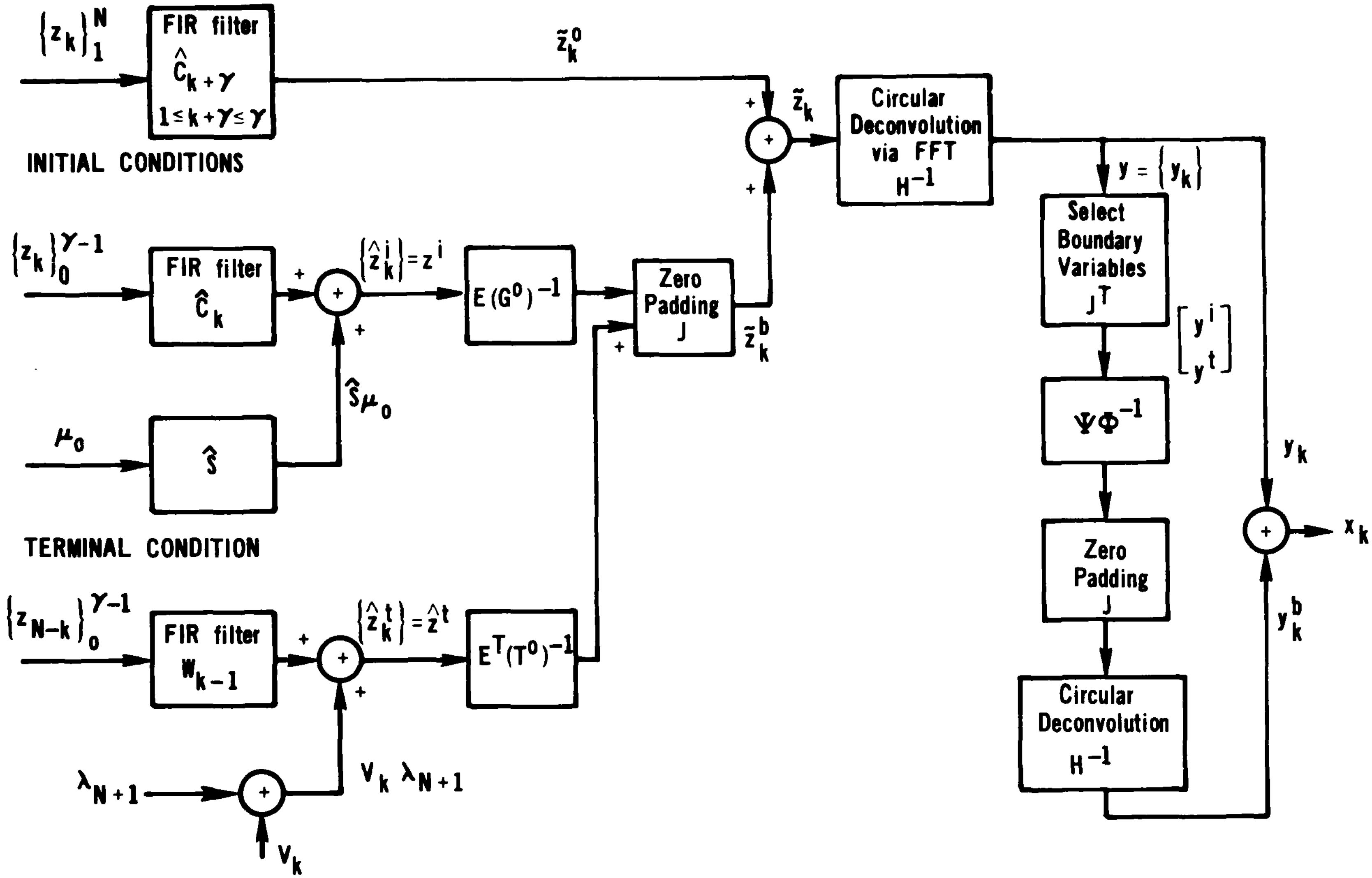


Figure 1. Solution of the Two Point Boundary Value Problem by Circular Decomposition and FFT. The Fixed Interval Smoothing Problem is Solved by Setting $\mu_0 = 0$, $\lambda_{N+1} = 0$.

(5.3) $$R_{N+1} = \underline{X}_{N+1} = A\underline{X}_N + BKB^T .$$

Thus, R_{N+1} is the response of a boundary value system excited by an impulse I_n at the terminal boundary. As a result, the foregoing algorithm for solving such boundary value problems can also be used to construct the Riccati matrix R_{N+1} columnwise.

Conceptually, it appears that n consecutive vector boundary value problems have to be solved. However, there are some significant simplifications which reduce the problem to solving one circular deconvolution problem of size O(N) and one matrix inversion of size $2m\gamma$. The steps 1 through 7 given above are replaced by the following steps obtained by setting $z_k = 0$ in the main algorithm. In the following we will use uppercase variables to signify the fact that X, Λ, etc. are now matrices.

Step 1: Extract the terminal variables

$$\hat{Z}_k^t = V_k I_n = V_k .$$

Step 2: Determine the terminal γ submatrices $\tilde{Z}^t$ of the block vector $\underline{\tilde{Z}}$

$$\tilde{Z}^t \triangleq E^T (T^0)^{-1} \hat{Z}^t .$$

Step 3: Compute the block vector $J^T\underline{Y}$ as

$$J^T\underline{Y} = J^T H^{-1} J \begin{bmatrix} 0 \\ \tilde{Z}^t \end{bmatrix}$$

Note that only the boundary values are needed in the sequel, hence direct convolution may be more efficient than circulant deconvolution via the FFT. The order of $J^T H^{-1} J$ is $(2m\gamma) \times (2m\gamma)$.

Step 4: Find the boundary values

$$B = \Phi^{-1} J^T \underline{Y}$$

where $\Phi = [I_{2m\gamma} - J^T H^{-1} J\Psi]$ as before.

Step 5: Recall that $B = \begin{bmatrix} \hat{X}^i \\ \hat{X}^t \end{bmatrix}$, so the last components $\tilde{X}^t$ are obtained from

$$\tilde{X}^t = (T^0)^{-1}T^1\hat{X}^t + (T^0)^{-1}\hat{Z}^t \quad \text{(see (4.21))}$$

Step 6: Using (4.21) and the elements of $\tilde{X}^t$ we can obtain $\underline{X}_N$, the state estimate at time k = N. This gives R_{N+1} via (5.3). This algorithm is illustrated in Figure 2. Steps 1, 2 and 3 above are equivalent to circular convolution of the matrix sequence $[\underline{0}, \underline{0}, \ldots, 0, \tilde{Z}_1^t, \tilde{Z}_2^t, \ldots, \tilde{Z}_\gamma^t]$ with $[H^{-1}]_k$ and extraction only the boundary values of the output Y_k. The FFT algorithm is needed only once to compute $[H^{-1}]_k$, which are the matrix block elements of the first column of the block circulant matrix H^{-1}. Assuming that the systems in steps 4 and 5 are solved by LU decomposition, and the LU decompositions of Φ and T^0 as well as the matrices T^1 and $J^TH^{-1}J$ are precomputed, the computational effort for obtaining R_{N+1} will be approximately $6n^3$, and does not depend on the number N which compares favorably with the direct recursive method (2.9) - (2.10) which has a complexity of $O(Nn^2)$ when A is in the given canonical form. The savings are even more dramatic, when the algorithm is used to compute $R_{\ell(N+1)}$, $\ell = 1, \ldots L$, since only the matrices Ψ and Φ will have to be updated at each step. We will elaborate on this point when we discuss a recursive block filter in the next paragraph.

(iv) Recursive Block Filtering

As an application of the ideas presented so far, we turn to a recursive block filter, i.e., a filter which smoothes one block of data at a time.

This will be an approximation to the global smoother, and can be used more readily, when data is only available in blocks, or when a time variant system is modeled by piecewise time invariant systems.

The block smoother operates just like the global smoother, except for the initial conditions. If the blocks are indexed by r, then the initial covariance for the block r is $R_{N+1,r-1}$, assuming the block size is N, where $R_{N+1,r-1}$ denotes

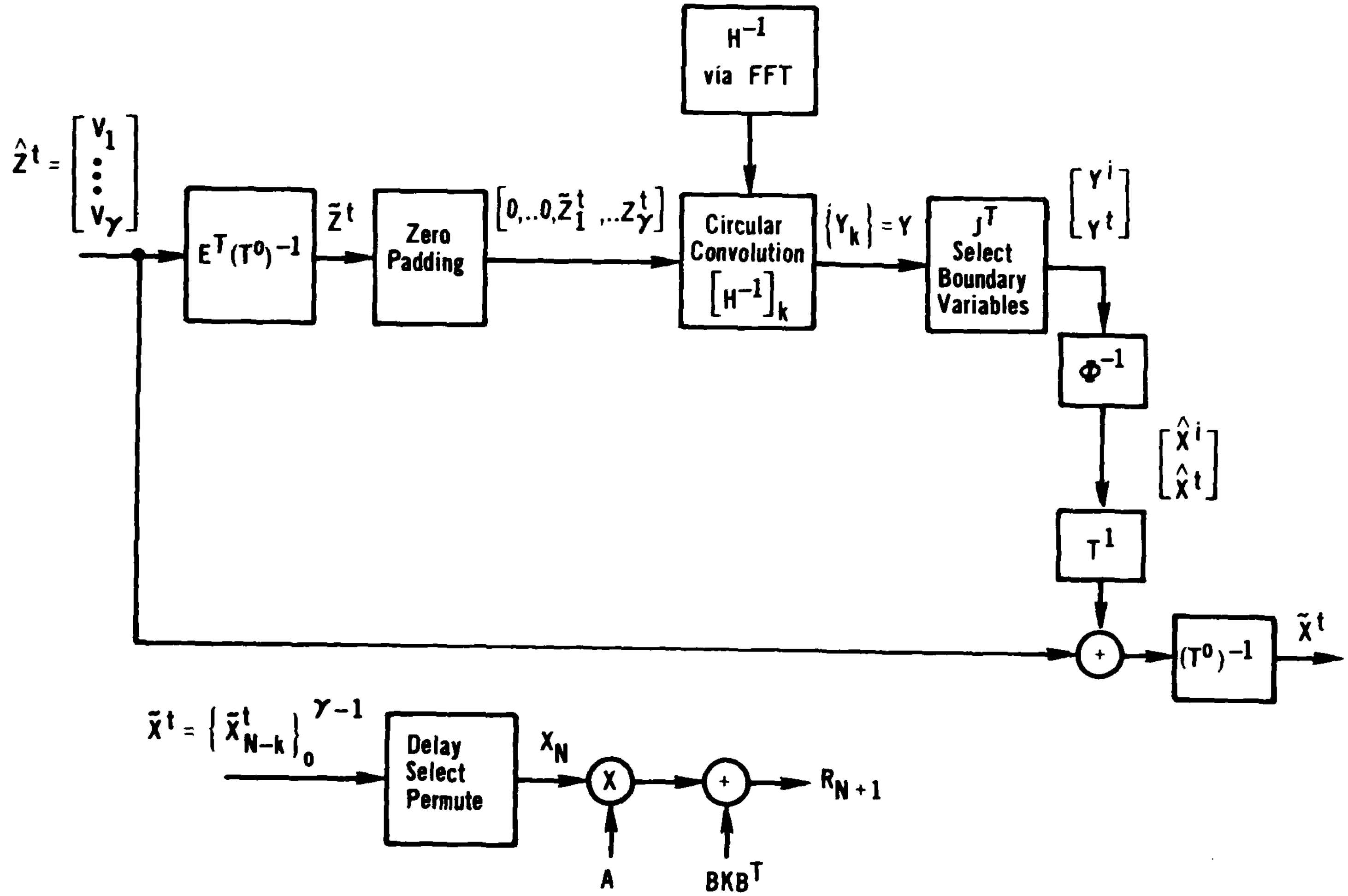


Figure 2. Solution of Riccati Equation at $k = N+1$ via the FFT

the Riccati matrix. The initial value $\hat{x}_{0,r}$ also has to include the one step predictor $s_{N+1,r-1}$ from the block (r-1). Thus, the block filter is of the form

(5.1) $$\underline{x}_{k+1,r} = A\underline{x}_{k,r} + BKB^T\underline{\lambda}_{k+1,r}$$

(5.2) $$\underline{\lambda}_{k,r} = A^T\underline{\lambda}_{k+1,r} + C^TR^{-1}[z_{k,r} - C\hat{\underline{x}}_{k,r}]$$

(5.3) $$\hat{\underline{x}}_{0,r} = Q_{0,r}\underline{\lambda}_{0,r} + \mu_r \; ; \; \underline{\lambda}_{N+1,r} = 0 \; ,$$

$$Q_{0,r} = R_{N+1,r-1}, \; \mu_r = s_{N+1,r-1}$$

This algorithm uses the two algorithms previously described recursively.

Step 1: Set $r = 1$, initialize the parameters by setting $Q_{0,1} = Q_0$, $s_{N+1,0} = 0$. For block r, process the data using the boundary value algorithm for (5.1)-(5.3).

Step 2: Find the solution $R_{N+1,r}$ of the Riccati equation at the end of the data block using algorithm (iii).

Step 3: Update the following matrices:

$$Q_{0,r+1} = R_{N+1,r}$$

$$\mu_{r+1} = s_{N+1,r} = A\underline{x}_{N,r}$$

then use $Q_{0,r+1}$ to update G^0, Φ, and Ψ.

Step 4: If all blocks are processed, stop, otherwise set $r = r+1$ and go to Step 2.

This algorithm is illustrated in Figure 3.

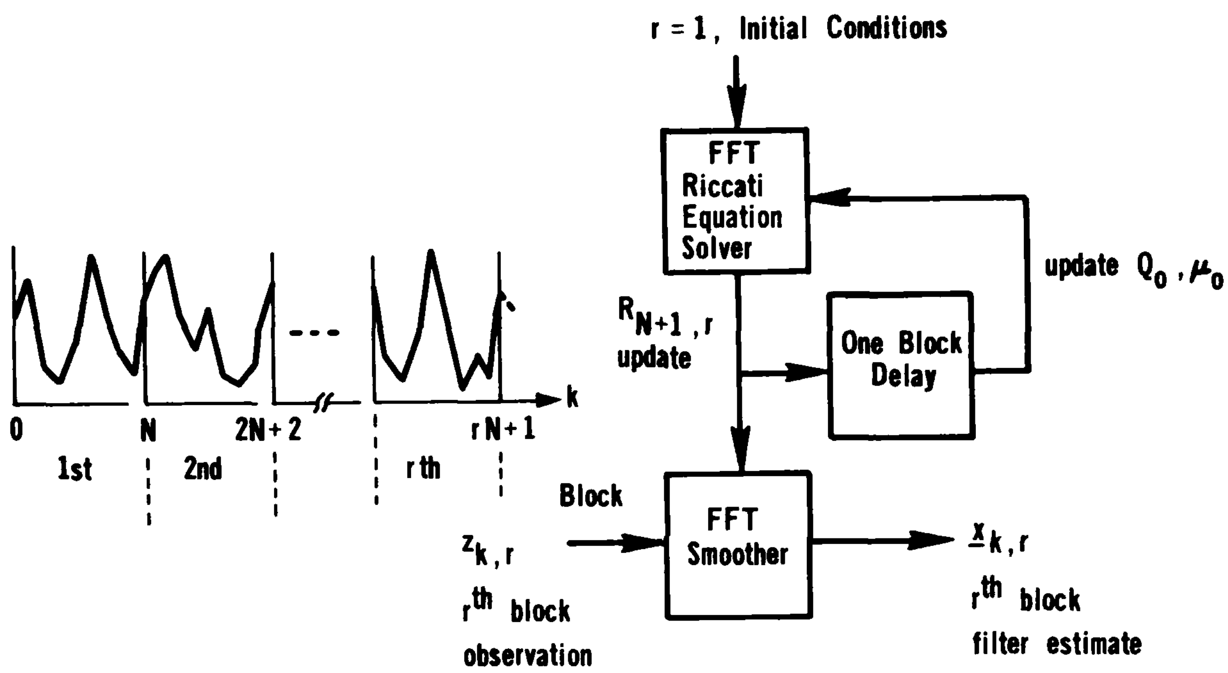


Figure 3. The Block Recursive Filter

6. REMARKS AND COMPUTATIONAL COMPLEXITY

a) We note that the above smoothing algorithm is nonrecursive and does not require the solution of a Riccati equation or equivalent recursive computations such as the solution of Toeplitz systems via Levinson-Trench type algorithms [24,25].

b) From Fig. 1 the smoothing filter output x_k can be written as

(6.1) $$x_k = y_k + y_k^b \triangleq x_k^o + x_k^b$$

(6.2) $$x_k^o \triangleq [H^{-1}\tilde{z}^o]_k, \quad x_k^b \triangleq [H^{-1}\tilde{z}^b]_k + y_k^b$$

The solution component $\{x_k^o\}$ could be considered as the FFT Wiener smoothing filter output, which is obtained by first sampling the Wiener smoothing filter output in the frequency domain followed by its inverse Fourier transform. The *frequency domain* Wiener filter equation is obtained by considering an infinite duration filter in the steady state. Specifically, the steady state, dynamic system equation can be written as

(6.3) $$\bar{x}_k - \sum_{\ell=1}^{\gamma} \bar{A}_\ell \bar{x}_{k-\gamma+\ell-1} = \varepsilon_{k-1}$$

(6.4) $$z_k = \sum_{\ell=1}^{\gamma} \bar{C}_\ell \bar{x}_{k-\gamma+\ell} + \eta_k$$

where $k\varepsilon(-\infty,\infty)$, and

(6.5) $$X(\omega) = \sum_{k=-\infty}^{\infty} x_k \exp(-jk\omega)$$

denotes the Fourier transform of $\{x_k\}$. Then the frequency domain Wiener filter estimate $X^o(\omega)$ for the smooth estimate of $\bar{X}_k$ is

(6.6) $$X^o(\omega) = [C^*(\omega)R^{-1}C(\omega) + S_x^{-1}(\omega)]^{-1}C^*(\omega)R^{-1}Z(\omega)$$

where $S_x(\omega)$ and R denote the power spectrum densities of x_k and η_k respectively, * denotes the conjugate transpose, and

$$(6.7) \qquad C(\omega) = \sum_{\ell=1}^{\gamma} \bar{C}_\ell \exp[-j(\gamma-\ell)\omega]$$

The quantity $S_x^{-1}(\omega)$ is obtained from the state equations (6.3), (6.4) as

$$(6.8) \qquad S_x^{-1}(\omega) = \sum_{k=1}^{\gamma+1} \sum_{\ell=1}^{\gamma+1} \bar{A}_\ell^T K^{-1} \bar{A}_k \exp[j(k-\ell)\omega]$$

Now defining

$$(6.9) \qquad H(\omega) = C^*(\omega)R^{-1}C(\omega) + S_x^{-1}(\omega), \; \tilde{Z}^o(\omega) = C^*(\omega)R^{-1}Z(\omega)$$

we can write (6.6) as

$$(6.10) \qquad H(\omega)X^o(\omega) = \tilde{Z}^o(\omega)$$

which gives the discrete Fourier transform of (4.12) at $\omega = \frac{2n\pi}{N-\gamma}$, $n = 0, \ldots, N-\gamma-1$.

Thus if we sample (6.6) at $\omega = \frac{2n\pi}{M}$, $n = 0, \ldots M-1$, i.e., let

$$\tilde{Z}(\omega) = \sum_{n=0}^{M-1} \hat{\tilde{z}}_n \delta(\omega - \frac{2n\pi}{M})$$

$$X^o(\omega) = \sum_{n=0}^{M-1} \hat{x}_n^o \delta(\omega - \frac{2n\pi}{M})$$

in (6.6), and take the inverse Fourier transform, we obtain the solution of the circular deconvolution problem defined by

$$x^o = H^{-1}\tilde{z} \quad \text{for} \quad M = N-\gamma$$

Moreover, it is easy to verify that $\{x_n^o\}$ and $\{\tilde{z}_n\}$ are the M step inverse discrete Fourier transforms of $\{x_k^o\}$ and $\{\tilde{z}_k\}$ respectively, and could be implemented via FFT.

c) If the system is not controllable, but remains observable, the above algorithm can be modified by working with a canonical observable model. Details are left to the reader.

d) The algorithm given above is based on an extension of a method of inversion of banded Toeplitz matrices to banded block-Toeplitz-matrices [29].

e) In the special case of single input systems, the dimensionality is significantly reduced. Moreover, the block circulant matrix H becomes a symmetric circulant matrix, and thus also the Fourier transform of the first row of H will be real, which reduces the number of operations.

Computational Complexity

We discuss separately the initial computational effort, which can be computed off-line. Secondly, we address the complexity of the data processing. Assume the input dimension is m, the state dimension is n, the output dimension is p.

Initial effort: (let $M = N-\gamma$)

(1)	Find K^{-1} and R^{-1} in	$m^3 + p^3$ op.
(2)	Find $K^{-1}A_j$, $R^{-1}C_j$ in	$n(m^2 + p^2)$ ops.
(3)	$A_k^TK^{-1}A_j$, $C_k^TR^{-1}C_j$	$(p^3 + m^3)\frac{\gamma(\gamma+1)}{2}$ ops.
(4)	γ matrices V_i are needed	$2\gamma m^3$ ops.
(5)	F requires	$2\gamma m^3$ ops.
(6)	Set up Ψ in	$\frac{4}{3}(m\gamma)^3 + (\frac{\gamma(\gamma+1)}{2})m^3 + \frac{4}{3}m\gamma^3 - \frac{m\gamma}{3}$

(7) Find the LU decomposition of $[I - J^T H^{-1} J\Psi]$	$\frac{(2m\gamma)^3}{3} - \frac{2m\gamma}{3}$
(8) Find H^{-1}	$2m^2 M\log M + Mm^3$
(9) $J^T H^{-1} J\Psi$	$8\gamma^3 m^3$ operations

Total: $M(2m^2\log M + m^3) + m^3(1 + \frac{32}{3}\gamma^3 + \frac{\gamma^2+9\gamma}{2}) + m^2 n + np^2 + \frac{p^3\gamma(\gamma+1)}{2} - m\gamma$

Data Processing

Steps 1,2	$Nnp + \gamma^2 mp$ ops.
Step 3	$4(m\gamma)^2$ ops.
Step 4	$2mM\log M + Mm^2$ ops.
Step 5	$2 \cdot (2m\gamma)^2$
Step 6	$2mM\log M + Mm^2$ ops.
Step 7	$4(m\gamma)^2$ ops.
Total:	$Nnp + 4mM\log M + 2Mm^2 + 16m^2\gamma^2$

These operations are for the controllable canonical form. Additional operations are required for transformation of the system parameters and state estimates if the given system is not in this form.

7. EXAMPLES

Example 1: To illustrate the general ideas of the derivation of the smoothing algorithm via FFT we consider the special case of (4.1) - (4.2) where $m = n$, $\gamma = 1$ and the covariances K, Q_0, and R are identity matrices.

(7.1) $$x_{k+1} = Ax_k + \lambda_{k+1}$$

(7.2) $$\lambda_k = A^T\lambda_{k+1} + C^T(z_k - Cx_k), \; k = 0, \dots N$$

with boundary conditions

$$x_0 = \lambda_0 \quad \text{and} \quad \lambda_{N+1} = 0 \ .$$

When (7.1) is solved for λ_{k+1} and the expression is substituted into (7.2) we obtain

(7.3) $$x_k - Ax_{k-1} = A^T(x_{k+1} - Ax_k) + C^T(z_k - Cx_k), \; k = 1, \dots N-1 \ .$$

We can collect terms to get

(7.4) $$-Ax_{k-1} + (I + A^TA + C^TC)x_k - A^Tx_{k+1} = C^Tz_k \ .$$

If we define

$$\tilde{A}_{-1} \triangleq -A \ , \ \tilde{A}_0 \triangleq (I + A^TA + C^TC) \ , \ \tilde{A}_1 = -A^T$$

then we can write the resulting system (7.4) for $k = 1, \dots N-1$ as

(7.5) $$\begin{bmatrix} \tilde{A}_{-1} & \tilde{A}_0 & \tilde{A}_1 & & 0 & \\ 0 & \ddots & \ddots & \ddots & \tilde{A}_1 & \\ & & \tilde{A}_{-1} & & \tilde{A}_0 & \tilde{A}_1 \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \tilde{z}_1^0 \\ \vdots \\ \tilde{z}_{N-1}^0 \end{bmatrix} , \text{ where } \tilde{z}_k^0 = C^Tz_k \ .$$

By putting the boundary terms involving x_0 and x_N on the right hand side we obtain the perturbed block Toeplitz system

(7.6)
$$\begin{bmatrix} \tilde{A}_0 & \tilde{A}_1 & & 0 \\ \tilde{A}_{-1} & \ddots & \ddots & \tilde{A}_1 \\ 0 & & \tilde{A}_{-1} & \tilde{A}_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1^o \\ \vdots \\ \tilde{z}_{N-1}^o \end{bmatrix} + \begin{bmatrix} -\tilde{A}_{-1}x_0 \\ 0 \\ \vdots \\ 0 \\ -\tilde{A}_1 x_N \end{bmatrix} .$$

Now x_0 and x_N can be eliminated by applying the boundary conditions.

The initial condition $x_0 = \lambda_0$ combined with (7.2) leads to

(7.7)
$$x_0 = A^T\lambda_1 + C^T(z_0 - Cx_0) .$$

(7.1) can be used to eliminate λ_1.

(7.8)
$$x_0 = A^T(x_1 - Ax_0) + C^T(z_0 - Cx_0).$$

Collecting terms we obtain the initial relationship

(7.9)
$$(I + A^TA + C^TC)x_0 = A^Tx_1 + C^Tz_0$$

which is clearly uniquely solvable for x_0 in terms of x_1 and z_0.

The terminal condition $\lambda_{N+1} = 0$ combined with (7.2) leads to

(7.10)
$$\lambda_N = C^T(z_N - Cx_N) .$$

(7.1) can be used to eliminate λ_N:

(7.11)
$$x_N - Ax_{N-1} = C^T(z_N - Cx_N) .$$

Collecting terms we obtain

(7.12)
$$(I + C^TC)x_N = Ax_{N-1} + C^Tz_N$$

which is again uniquely solvable for x_N in terms of x_{N-1} and z_N.

Thus (7.9) and (7.12) can be used to eliminate x_0 and x_N in (7.6): let $G_0 \triangleq I + A^TA + C^TC$ and $T_0 \triangleq I + C^TC$, then

$$(7.13)\quad \begin{bmatrix} \tilde{A}_0 & \tilde{A}_1 & & 0 \\ \tilde{A}_{-1} & \ddots & \ddots & \\ & \ddots & \ddots & \tilde{A}_1 \\ 0 & & \tilde{A}_{-1} & \tilde{A}_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} \tilde{z}_1^o \\ \vdots \\ \tilde{z}_{N-1}^o \end{bmatrix} + \begin{bmatrix} -\tilde{A}_{-1}G_0^{-1}(A^T x_1 + C^T z_0) \\ 0 \\ \vdots \\ 0 \\ -\tilde{A}_1 T_0^{-1}(Ax_{N-1} + C^T z_N) \end{bmatrix}$$

Finally, this problem can be rewritten as a perturbed block circulant system:

$$(7.14)\quad H\begin{bmatrix} x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \underline{\tilde{z}} + J\Psi \begin{bmatrix} x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}, \text{ where}$$

$$H = \begin{bmatrix} \tilde{A}_0 & \tilde{A}_1 & 0 & 0 & \tilde{A}_{-1} \\ \tilde{A}_{-1} & \ddots & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \tilde{A}_1 \\ \tilde{A}_1 & 0 & 0 & \tilde{A}_{-1} & \tilde{A}_0 \end{bmatrix}, \quad \underline{\tilde{z}} = \tilde{z}^o + \begin{bmatrix} -\tilde{A}_{-1}G_0^{-1}C^T z_0 \\ 0 \\ \vdots \\ 0 \\ -\tilde{A}_1 T_0^{-1} C^T z_N \end{bmatrix}$$

$$J = \begin{bmatrix} I_m & 0 \\ 0 & 0 \\ 0 & I_m \end{bmatrix}, \quad \Psi = \begin{bmatrix} -\tilde{A}_{-1}G_0^{-1}A^T & \tilde{A}_{-1} \\ \tilde{A}_1 & -\tilde{A}_1 T_0^{-1} A \end{bmatrix}$$

letting $b = \begin{bmatrix} x_1 \\ x_{N-1} \end{bmatrix}$ we can first solve for the boundary term b as

$$(7.15)\quad b = J^T H^{-1}\underline{\hat{z}} + J^T H^{-1} J\Psi b \ .$$

This is a system of order 2m. Finally the components $x_2 \ldots x_{N-2}$ can be obtained via FFT as

$$\begin{bmatrix} x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = H^{-1}(\underline{\tilde{z}} + J\Psi b) \ .$$

Example 2 (A detailed solution of Riccati Equation)

Suppose the Riccati matrix R_8 has to be evaluated for the following systems parameters:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1.2 & -1.3 \\ -2.1 & -2.2 & -2.3 \end{bmatrix} \qquad K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; R = 1$$

$$C = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \; Q_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

i.e., this is a system with two inputs and one output. Note $\gamma = 2$ here.

We obtain

$$\tilde{A}_0 = \begin{bmatrix} 13.69 & 6.62 \\ 6.62 & 8.98 \end{bmatrix}, \; \tilde{A}_1 = \begin{bmatrix} 7.02 & 9.33 \\ 1.3 & 2.30 \end{bmatrix}, \; \tilde{A}_2 = \begin{bmatrix} 1 & 2.1 \\ 0 & 0 \end{bmatrix}$$

$$G^0 = \begin{bmatrix} 7.41 & 0 & 5.82 & 7.13 \\ 0 & 1 & 0 & 0 \\ 5.82 & 0 & 13.69 & 6.62 \\ 7.13 & 0 & 6.62 & 8.98 \end{bmatrix}$$

$$G^1 = \begin{bmatrix} -1 & -2.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -7.02 & -9.33 & -1 & -2.1 \\ -1.3 & -2.3 & 0 & 0 \end{bmatrix}$$

$$T^0 = \begin{bmatrix} 1.2 & 1.3 & 1 & 0 \\ 3.2 & 2.3 & 0 & 2 \\ -.2 & 0 & 0 & -1.2 \\ -2.3 & 2 & 0 & -2.3 \end{bmatrix}$$

$$T^1 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & -2.1 & 0 \\ -1 & 0 & -1.2 & -1.3 \\ -2.1 & 0 & -3.2 & -2.3 \end{bmatrix}$$

The first block row of the block circulant matrix H^{-1} is obtained as

$$H^{-1}_{1,\cdot} = \begin{bmatrix} .594 & -.181 & -.124 & -.479 & .059 & .081 & .059 & -.077 & -.124 & -.003 \\ -.181 & .656 & -.003 & .134 & -.077 & .045 & .081 & .045 & -.479 & .134 \end{bmatrix}$$

PHI

.983	-.008	-.004	-.009	.071	0.000	-.202	-.031
.024	1.029	.001	.002	-.119	0.000	-.752	-.051
.050	.041	1.007	.014	-.000	0.000	.152	.022
-.008	-.009	.001	1.001	.001	0.000	.142	.089
.256	.188	.024	.051	1.070	0.000	.105	.091
-.003	.058	-.002	-.004	.020	1.000	.058	.035
-.714	-.897	-.114	-.239	-.160	0.000	.811	-.199
.143	.219	.064	.134	-.069	0.000	-.093	.912

PSI

4.462	5.639	.704	1.478	1.000	0.000	7.020	1.300
5.639	7.232	.875	1.837	2.100	0.000	9.330	2.300
.704	.875	.116	.243	0.000	0.000	1.000	0.000
1.478	1.837	.243	.511	0.000	0.000	2.100	0.000
1.000	2.100	0.000	0.000	1.057	0.000	2.766	1.217
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7.020	9.330	1.000	2.100	2.766	0.000	10.471	3.153
1.300	2.300	0.000	0.000	1.217	0.000	3.153	1.429

Finally,

$$R_8 = \begin{bmatrix} 2.302 & -3.903 & -6.789 \\ -3.903 & 8.948 & 13.787 \\ -6.789 & 13.787 & 24.976 \end{bmatrix}$$

Example 3: (Computation of a Riccati Matrix)

This example illustrates the advantage of the FFT method over the two sweep method with respect to roundoff propagation. The systems parameters are

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 12 & -5 & 12 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix},$$

$$K = \begin{bmatrix} 1 & 0 \\ 0 & .49 \end{bmatrix}, \quad R = .16, \quad Q_0 = \begin{bmatrix} .05 & 0 & 0 \\ 0 & .01 & 0 \\ 0 & 0 & .09 \end{bmatrix}$$

This is an unstable dual input single output system.

a) When the FFT method is applied with N = 7 we obtain

$$R_8 = \begin{bmatrix} 2.3035 & 1.5797 & -10.807 \\ 1.5797 & 2.7446 & -6.2513 \\ -10.807 & -6.2513 & 70.4289 \end{bmatrix}$$

$R_{16} \approx R_8$, i.e., this is approximately the steady state solution.

b) The two sweep method gives

$$R_3 = \begin{bmatrix} 1.988 & 1.307 & -9.250 \\ 1.307 & 2.248 & -4.667 \\ -9.250 & -4.667 & 67.520 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 2.166 & 1.494 & -10.156 \\ 1.492 & 2.677 & -5.825 \\ -10.175 & -5.821 & 67.380 \end{bmatrix}$$

$$R_8 = \begin{bmatrix} 2.018 & 1.531 & -9.788 \\ 6.157 & 3.538 & -24.617 \\ 41.746 & 3.024 & -139.93 \end{bmatrix}$$

R_{13} contains elements of order 10^6, which shows the divergence of this method due to instability.

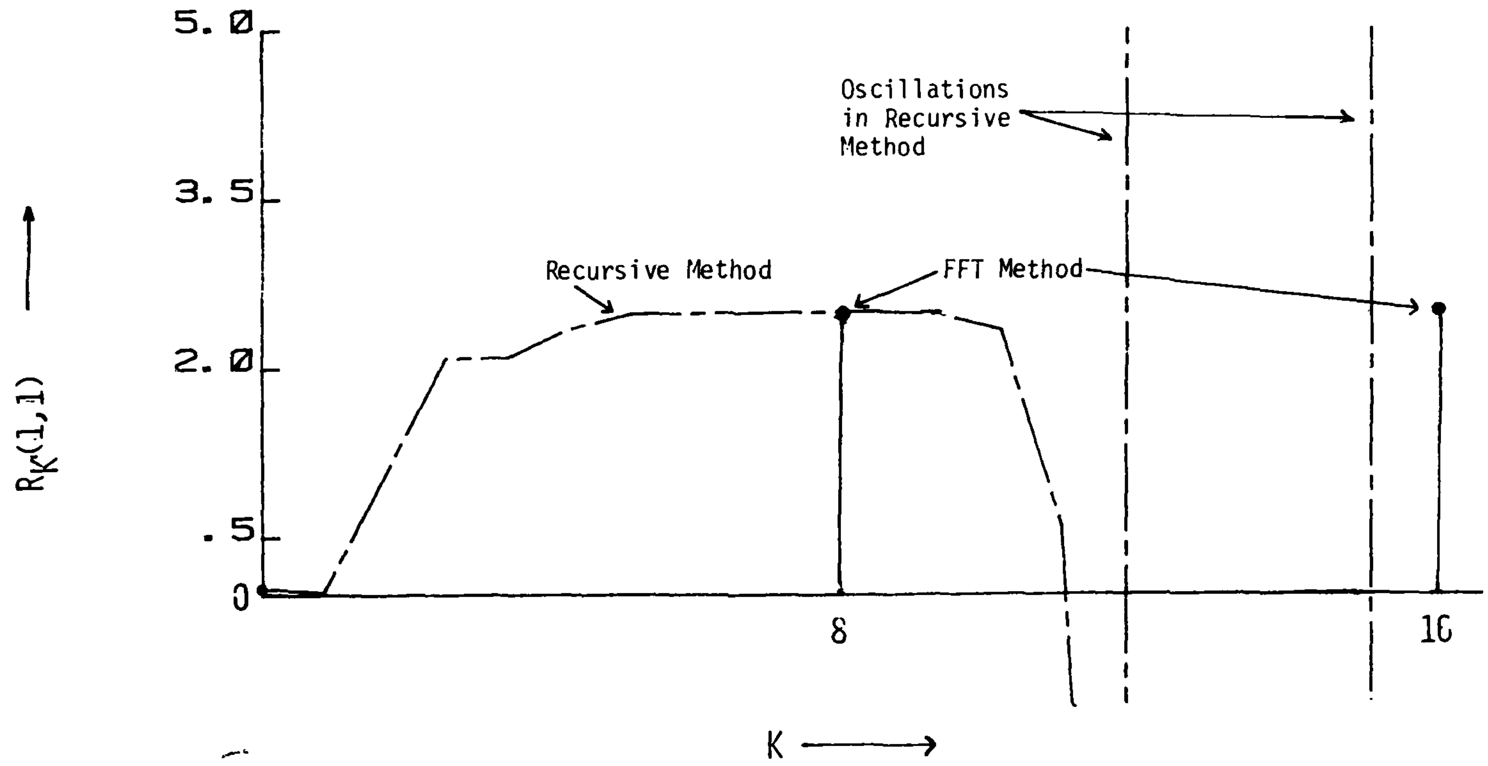


Figure 4: Riccati Equation Solution via the FFT Method. This example shows the numerical stability of the FFT algorithm.

Figure 4 shows the evolution of the element R_k (1,1). The equally spaced dashes show the sampled values using the FFT approach, and the other line shows the unstable behavior of the recursive method. Note that the line had to be clipped.

Example 4: (Smoothing and Recursive Block Filtering)

The following two examples illustrate solutions to the linear smoothing and recursive block filter problems. The performance of the filter is compared for different block sizes. The following system parameters were chosen:

Example 4.1: m = 1, n = 2, p = 2

$$A = \begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix},\ C = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix},\ Q_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ K = .01,\ R = \begin{pmatrix} .01 & 0 \\ 0 & .04 \end{pmatrix}$$

(see figures 5-8).

Example 4.2: m = 1, n = 2, p = 2

$$A = \begin{pmatrix} 0 & 1 \\ -.1 & -.2 \end{pmatrix},\ C = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix},\ Q_0 = \begin{pmatrix} .36 & 0 \\ 0 & .49 \end{pmatrix},\ K = 1,\ R = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

(see figures 9-12).

Note that the variances of the noises are considerably higher in Example 4.2.

For both examples the results are presented in the same format. The solid curves display the exact values of the states $x_k = \underline{x}_k(2)$, and the dotted curves refer to the estimates. Thirty data points are shown in each case. The following table contains the pertinent information. MSE refers to the mean square error between the exact and estimated values. Figures 7, 8 and 10, 11 are recursive block filters with different block sizes. Notice that the block filter estimates are quite close to the larger interval smoothing filter estimates.

Example 4.1

Figure	number of observations used	number of blocks	block size	MSE
5	51	1	51	.4497
6	30	1	30	.7448
7	30	2	15	1.60
8	30	3	10	1.099

Figure	number of observations used	number of blocks	block size	MSE
9	51	1	51	3.711
10	30	1	30	3.811
11	30	2	15	4.872
12	30	3	10	4.115

Example 5: (Sampling of a Riccati Matrix)

This example shows the evolution of the Riccati matrix R_k for the following systems parameters: $m = 1$, $n = 2$, $p = 1$

$$A = \begin{pmatrix} 0 & 1 \\ -.97 & -1.94 \end{pmatrix},\ Q_0 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},\ C = (1,1),\ B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ R = .2,\ K = .1.$$

Figures 13-15 display the components of R_k using the two sweep method as well as the sampled values using the FFT method. Note that the FFT method gives the same values as the recursive method at the sampled instants.

Conclusions

Fast nonrecursive algorithms for solving the discrete time fixed interval smoothing problem have been presented along with extensions to determine samples of the associated Riccati matrices at fixed times. Both algorithms have proven to be useful in implementing a block recursive filter to process data in batches.

The advantage of these algorithms does not merely lie in a reduction of the computational complexity compared to more conventional approaches. Their nonrecursive structure allows parallel architectures and the use of FFT firmware which results in faster execution times, and possibly a reduction in the accumulated roundoff error.

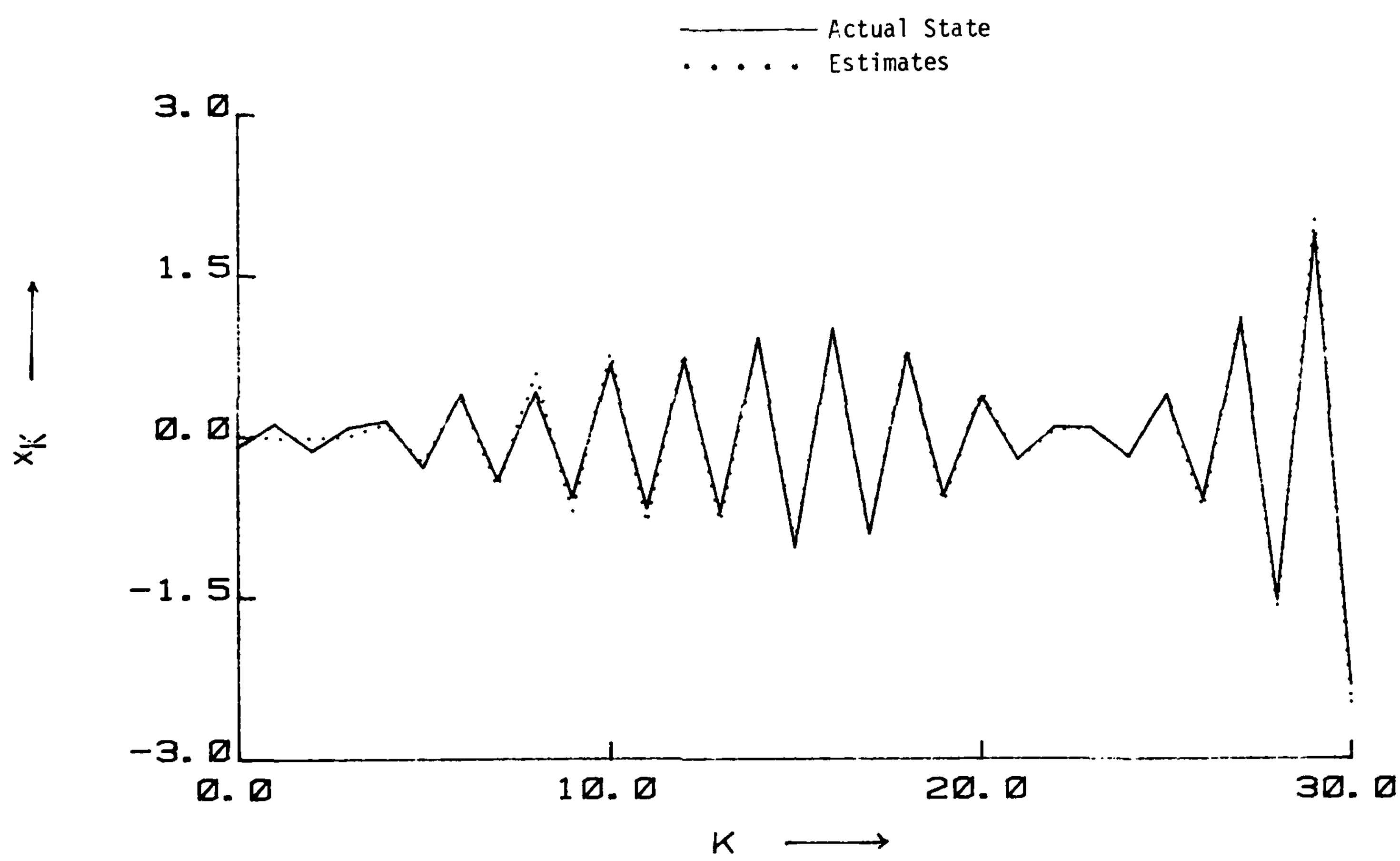


Figure 5: Fixed interval smoothing, Example 4.1, Block Size 51, only 31 samples are displayed.

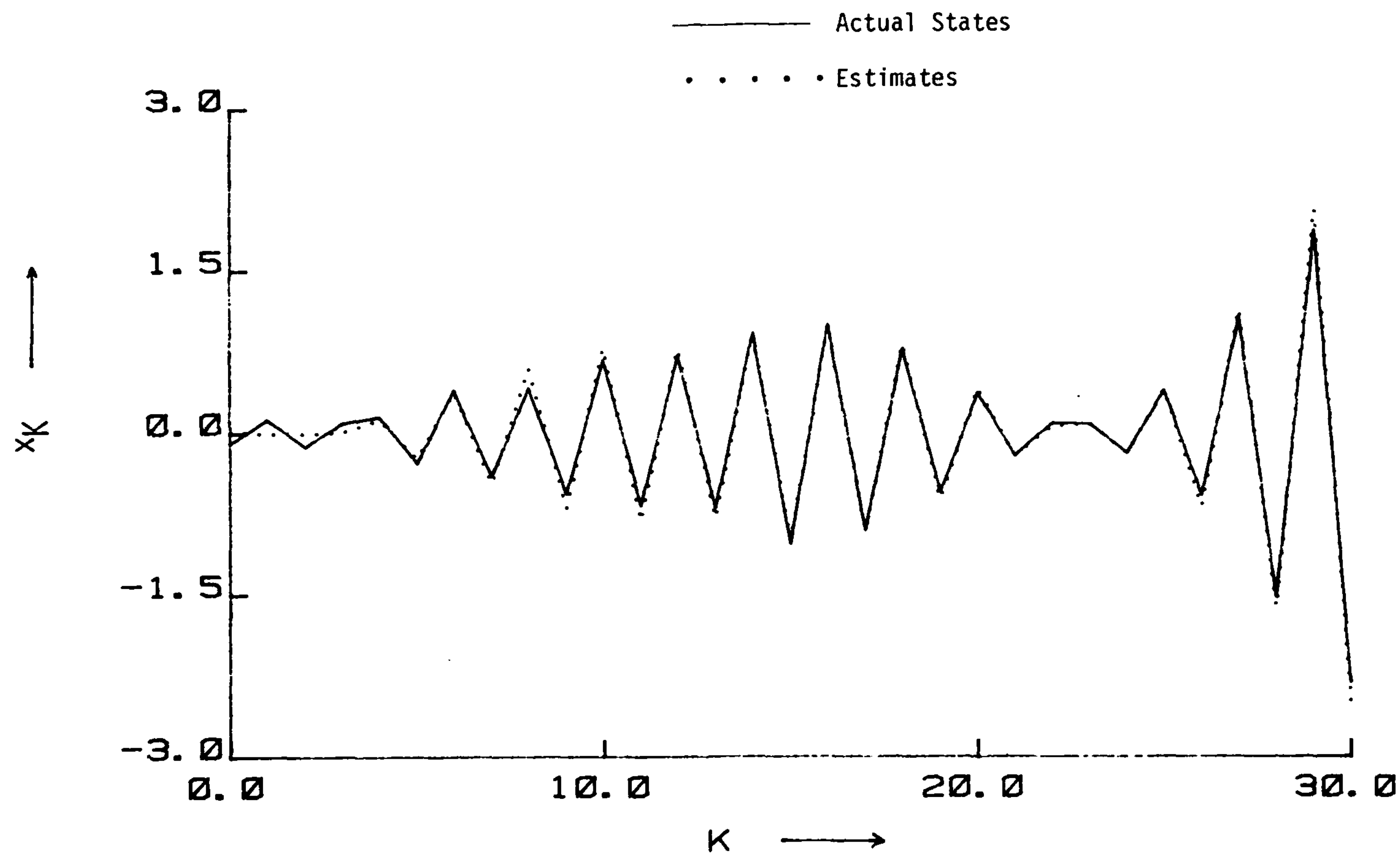


Figure 6: Fixed Interval Smoothing, block size 30, Example 4.1.

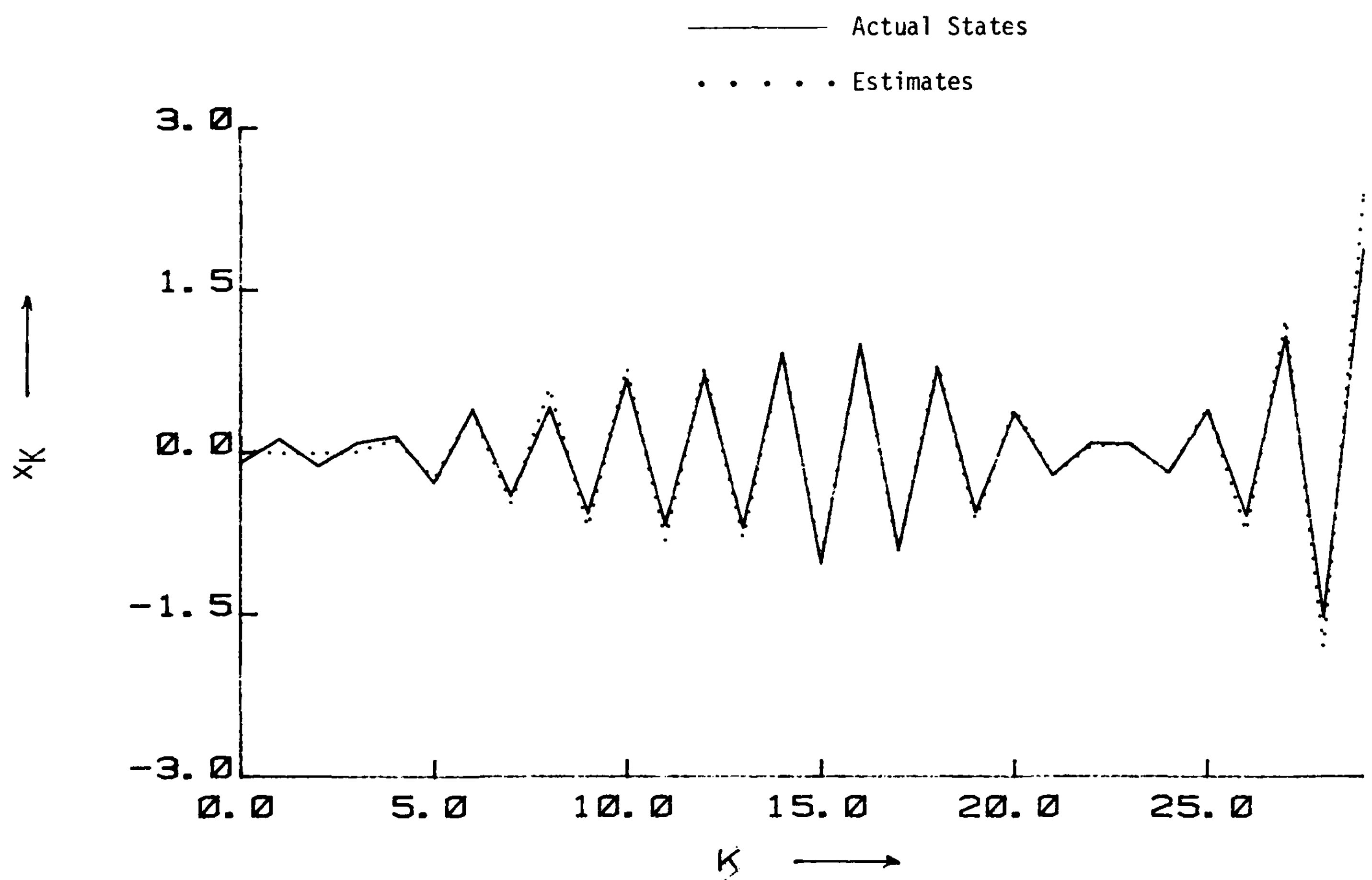


Figure 7: Recursive Block Filter, block size 15, Example 4.1

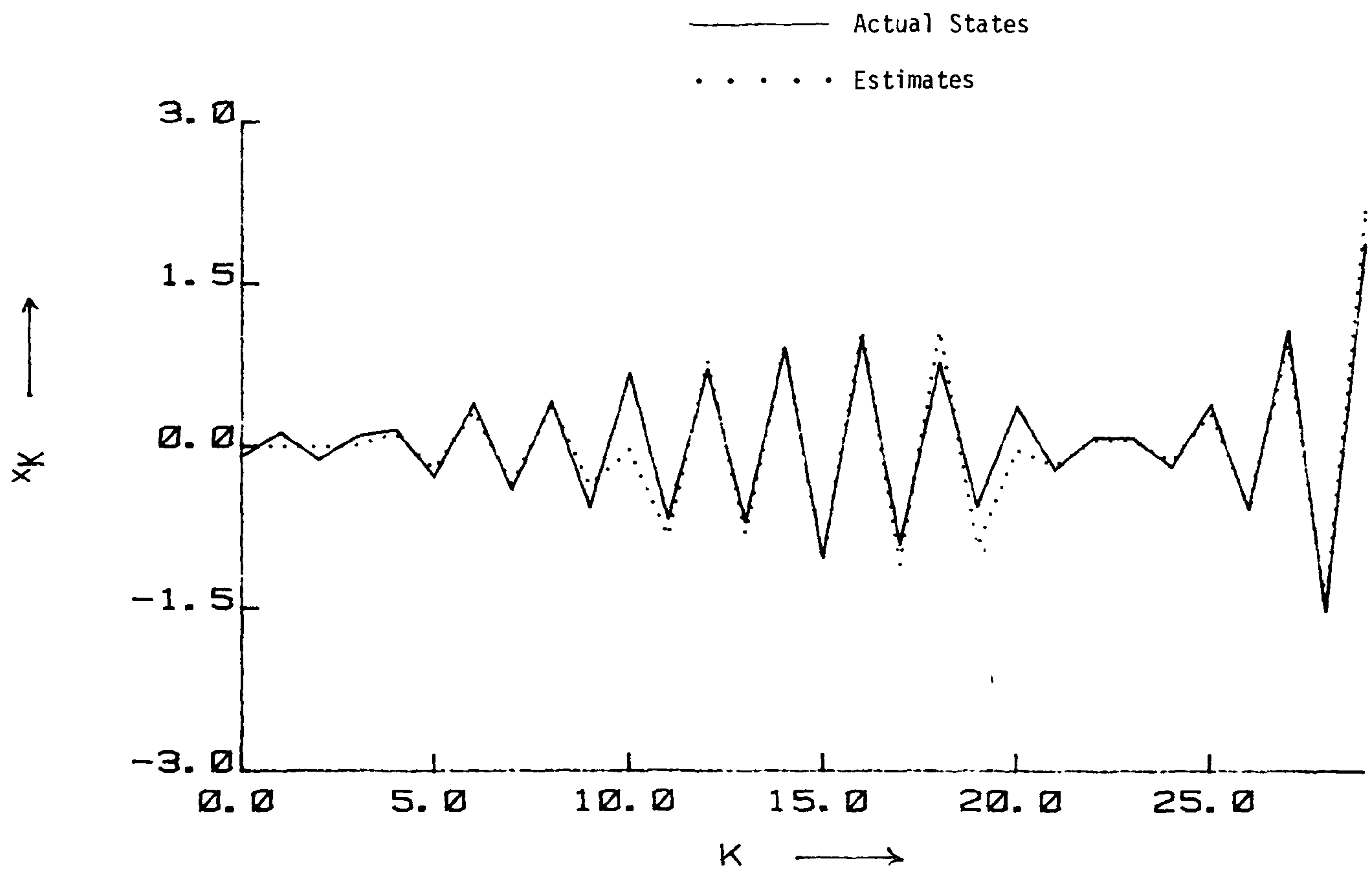


Figure 8: Recursive Block Filter, block size 10, Example 4.1

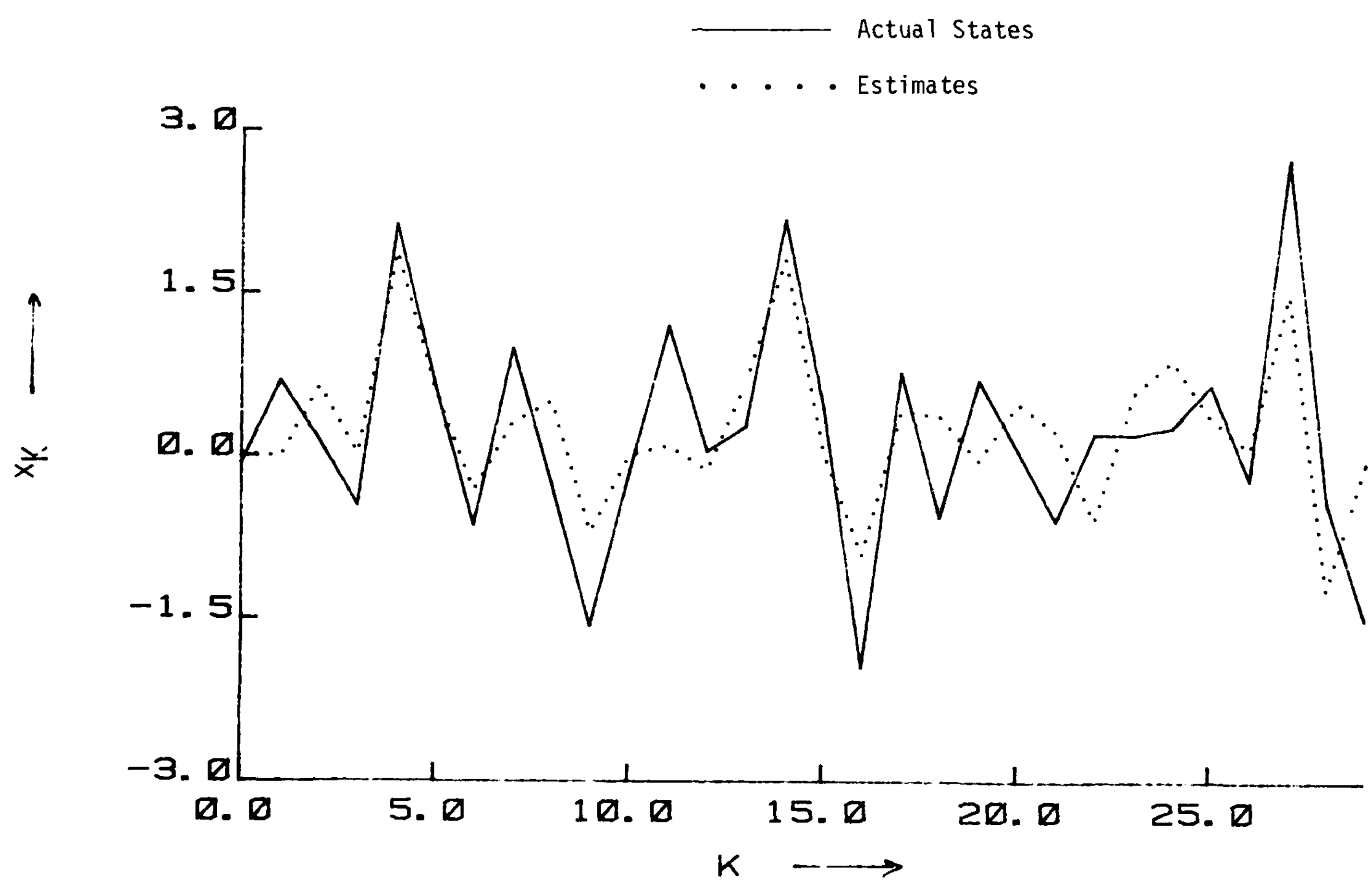


Figure 9: Fixed Interval Smoothing, block size 51, only 31 samples are displayed, Example 4.2

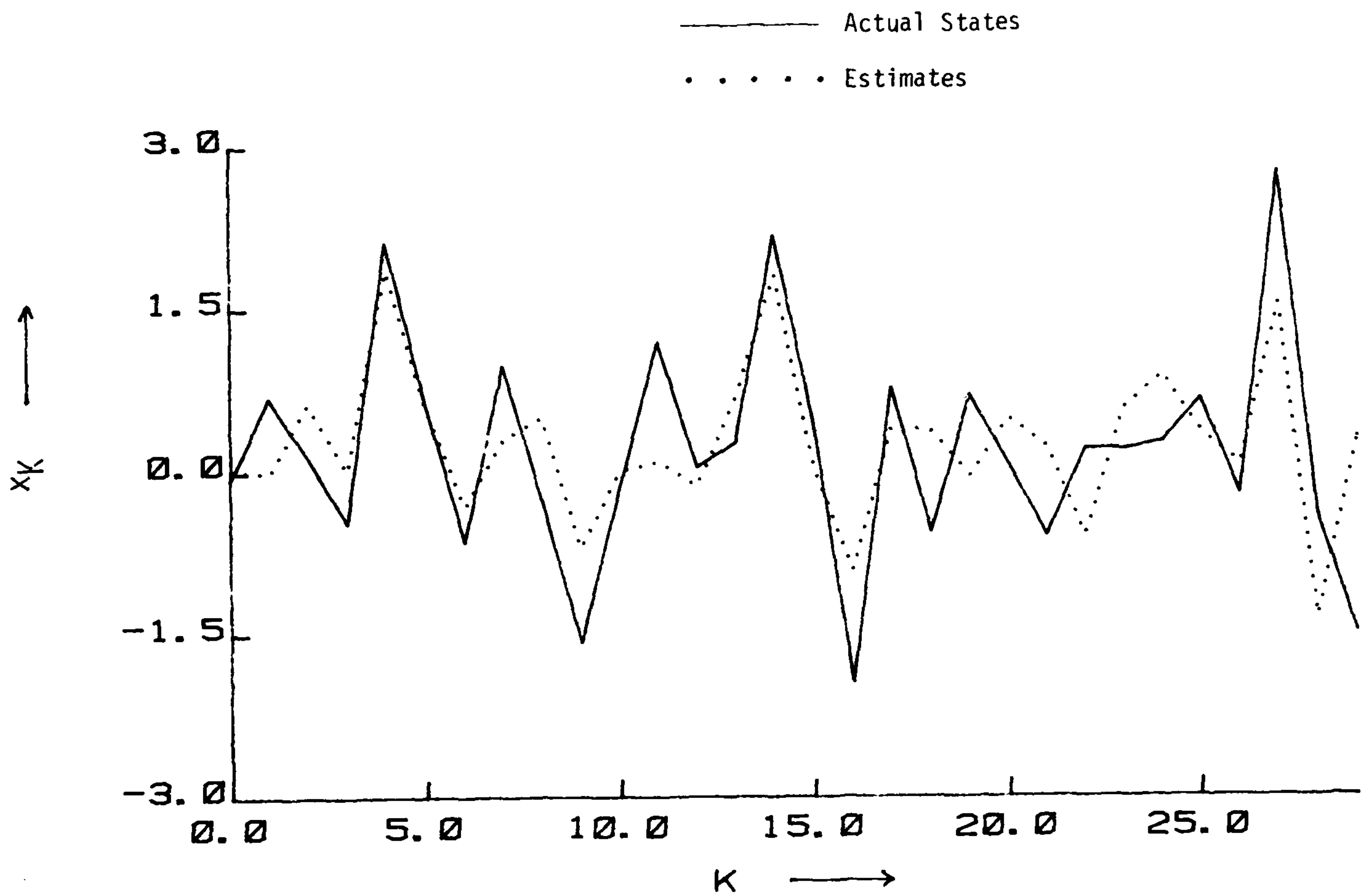


Figure 10: Fixed Interval Smoothing, block size 30, Example 4.2

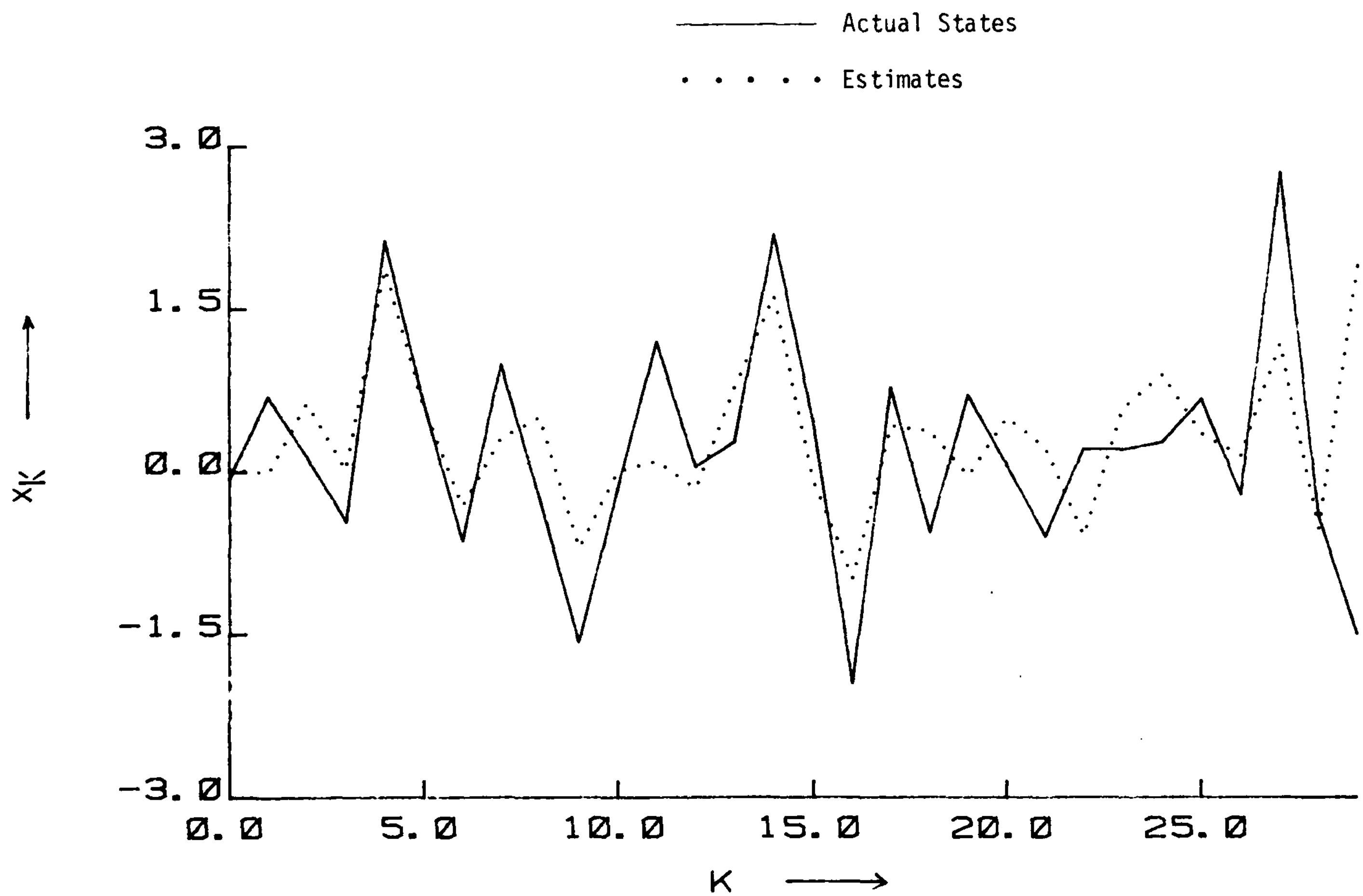


Figure 11: Recursive Block Filtering, block size 15, Example 4.2

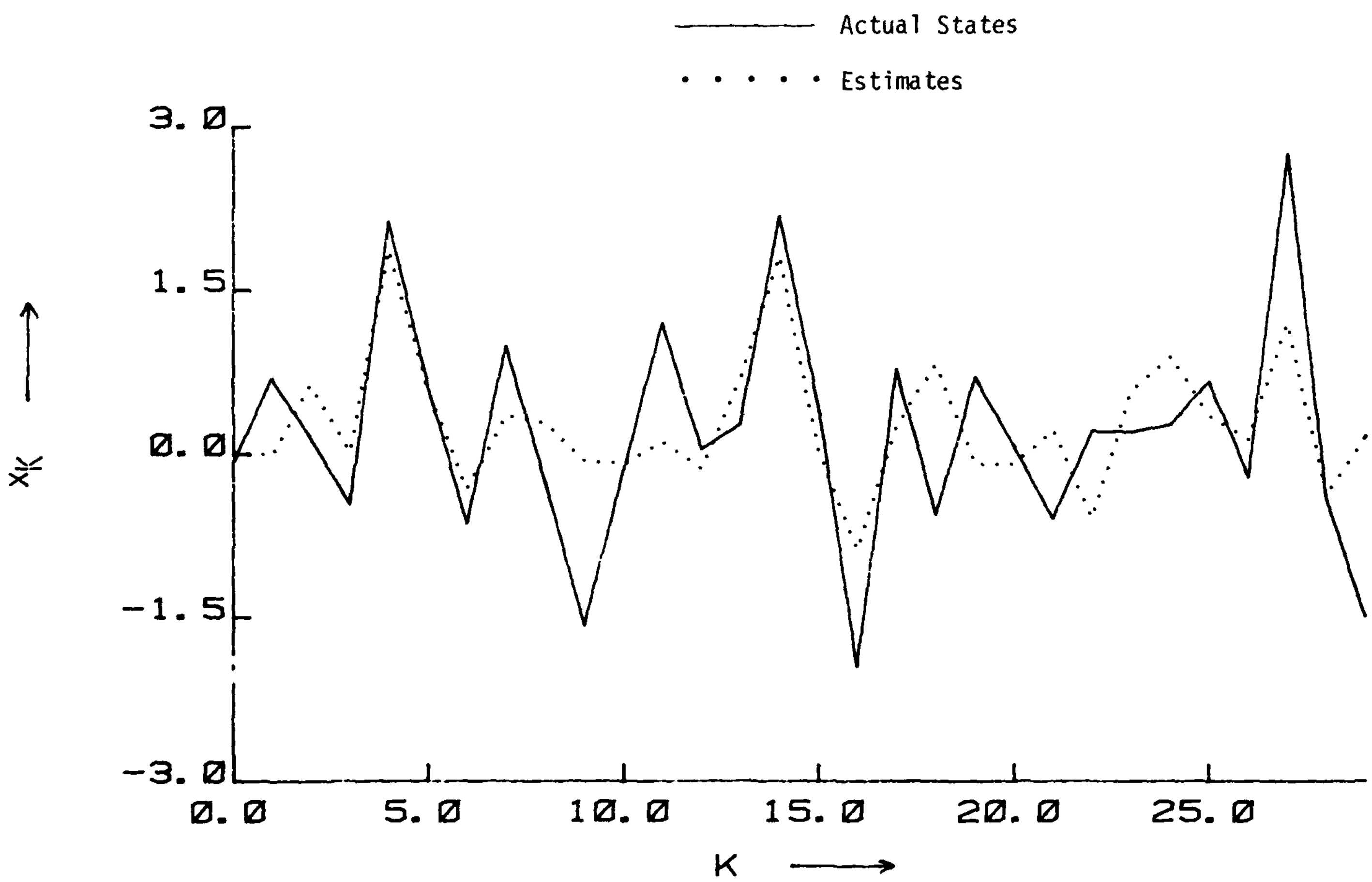


Figure 12: Recursive Block Filter, block size 10, Example 4.2

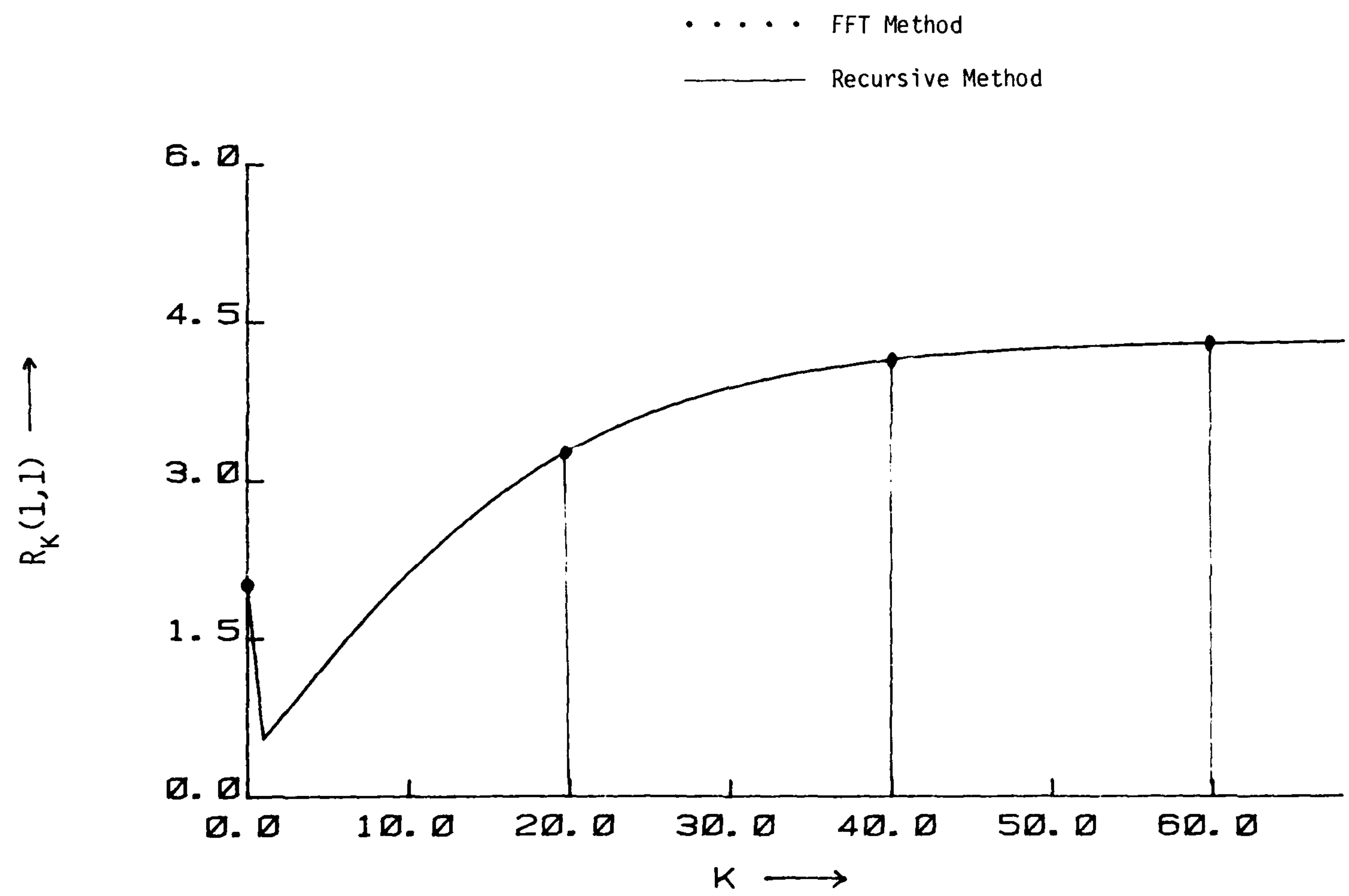


Figure 13: Riccati Equation Solutions

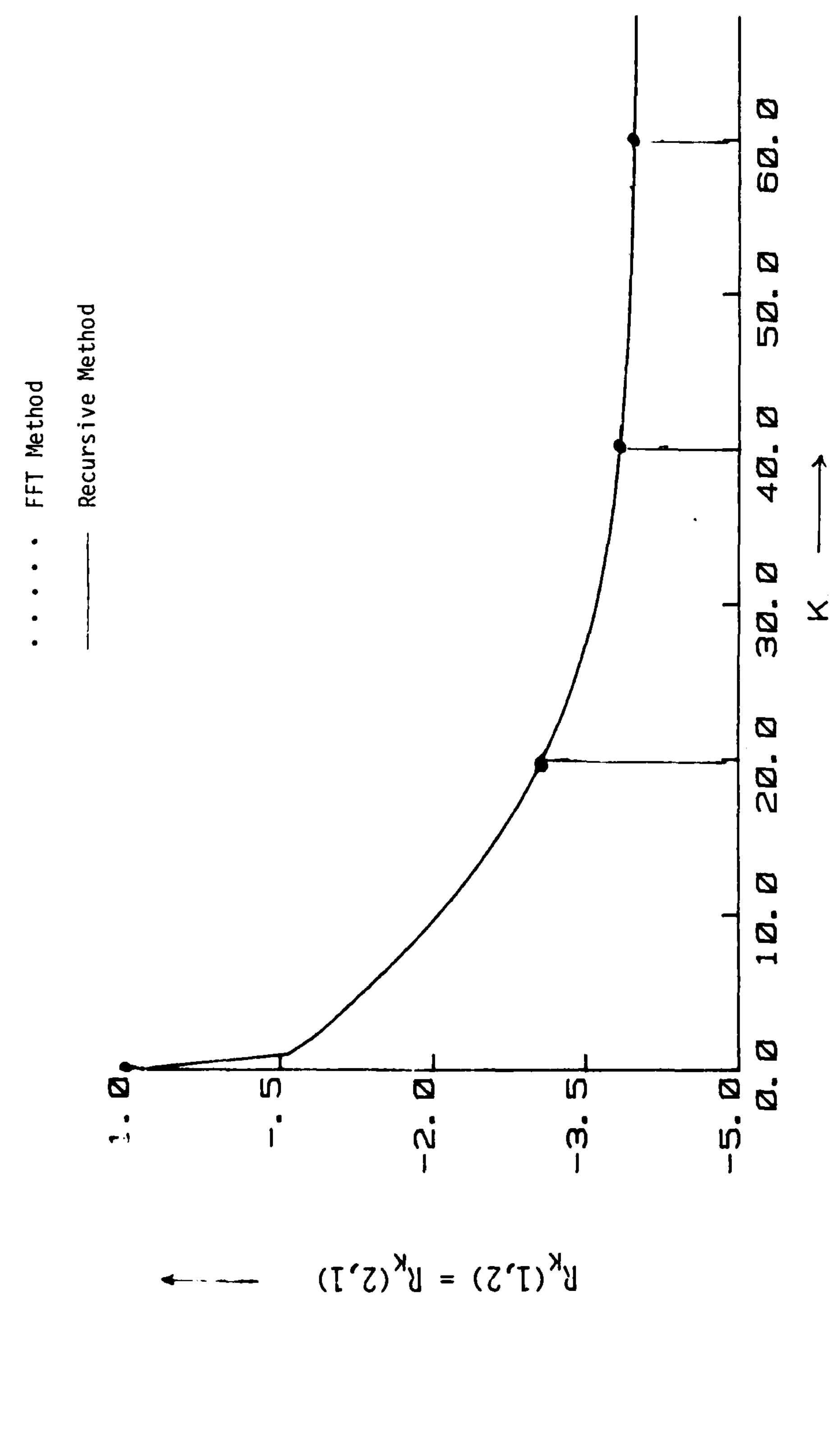


Figure 14: Riccati Equation Solutions

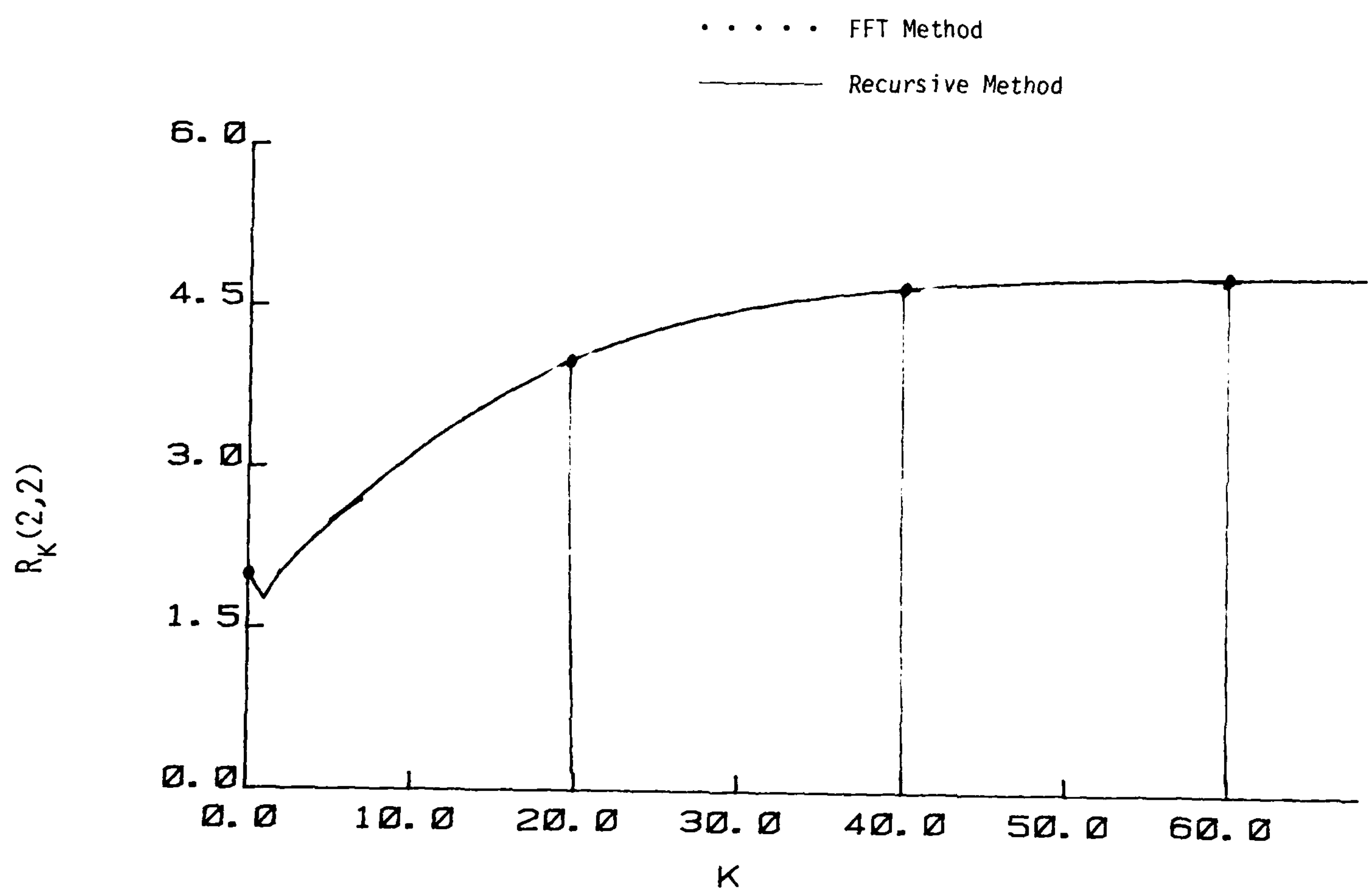


Figure 15: Riccati Equation Solution

APPENDIX A

A.1 A Technical Lemma:

Lemma (Change of Indices of Summation over a Rectangular Grid):

$$\text{(A1)}\quad \sum_{j=j_1}^{j_2}\sum_{i=i_1}^{i_2} a_{ij} = \sum_{\ell=i_1+j_1}^{i_2+j_1-1}\ \sum_{j=j_1}^{\ell-i_1} a_{\ell-j,j} + \sum_{\ell=i_2+j_1}^{j_2+i_1-1}\ \sum_{j=\ell-i_2}^{\ell-i_1} a_{\ell-j,j} + \sum_{\ell=i_1+j_2}^{i_2+j_2}\ \sum_{j=\ell-i_2}^{j_2} a_{\ell-j,j}\ ,$$

provided $(j_2-j_1) \geq (i_2-i_1)$, where $\ell = i+j$.

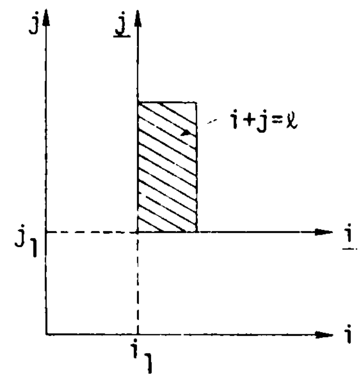


Figure 16

Proof For simplicity let $\underline{i} = i-i_1$ and $\underline{j} = j-j_1$. Then by assumption $\underline{j}_2 \geq \underline{i}_2$. Let $\underline{\ell} = \underline{i}+\underline{j}$. (See Figure 16).

Then

for $\underline{\ell} \leq \underline{i}_2$: $0 \leq \underline{j} \leq \underline{\ell}$

for $\underline{i}_2 \leq \underline{\ell} \leq \underline{j}_2$: $\underline{\ell}-\underline{i}_2 \leq \underline{j} \leq \underline{\ell}$

for $\underline{\ell} \geq \underline{j}_2$: $\underline{\ell}-\underline{i}_2 \leq \underline{j} \leq \underline{j}_2$

describes the range for $\underline{j}$. Note that in the special case $\underline{i}_2 = \underline{j}_2$ the middle formula is redundant. We now have to find the corresponding limits in terms of i,j and $\ell = i+j$. Since $\ell = \underline{\ell}+i_1+j_1$, or $\underline{\ell} = \ell-i_1-j_1$ these conditions become:

For $\ell-i_1-j_1 \leq i_2-i_1$: $0 \leq j-j_1 \leq \ell-i_1-j_1$

For $i_2-i_1 \leq \ell-i_1-j_1 \leq j_2-j_1$: $\ell-i_1-j_1-(i_2-i_1) \leq j-j_1 \leq \ell-i_1-j_1$

For $\ell-i_1-j_1 \geq j_2-j_1$: $\ell-i_1-j_1-(i_2-i_1) \leq j-j_1 \leq j_2-j_1$

This can be simplified to:

For $\ell \leq j_1+i_2$: $j_1 \leq j \leq \ell-i_1$

For $i_2+j_1 \leq \ell \leq j_2+i_1$: $\ell-i_2 \leq j \leq \ell-i_1$

For $\ell \geq j_2+i_1$: $\ell-i_2 \leq j \leq j_2$

Again, if $i_2-i_1 = j_2-j_1$, the middle formula is redundant. This proves the result and also its following corrollary:

Corrollary: if $i_2-i_1 = j_2-j_1$, then

$$\text{(A2)} \qquad \sum_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} a_{ij} = \sum_{\ell=i_1+j_1}^{i_1+j_2} \sum_{j=j_1}^{\ell-i_1} a_{\ell-j,j} + \sum_{\ell=i_1+j_2+1}^{i_2+j_2} \sum_{j=\ell-i_2}^{j_2} a_{\ell-j,j}$$

A.2 Initial Conditions:

Lemma:
$$\underline{\lambda}_k^1 = A_1^T\lambda_{k+1} + f_k^1$$

(A3)
$$\underline{\lambda}_k^i = A_i^T\lambda_{k+1} + f_k^i + \sum_{j=1}^{i-1}\left(\prod_{\ell=i-j}^{i-1} P_\ell^T\right)(A_{i-j}^T\lambda_{j+k+1} + f_{k+j}^{i-j})$$

Proof: The case i=1 follows from (4.2). Then use induction. For i=2 we obtain from (4.2):

$$\underline{\lambda}_k^2 = A_2^T\lambda_{k+1} + P_1^T\underline{\lambda}_{k+1}^1 + f_k^2$$

$$= A_2^T\lambda_{k+1} + P_1^TA_1^T\lambda_{k+2} + P_1^Tf_{k+1}^1 + f_k^2 \ ,$$

which is the claim for i=2. Now assume the result is correct for i. Then from (4.2) we find

$$\underline{\lambda}_k^{i+1} = P_i^T\underline{\lambda}_{k+1}^i + A_{i+1}^T\lambda_{k+1} + f_k^{i+1}$$

By the induction hypothesis we can replace λ_{k+1}^i. It suffices to show that

$$P_i^T\underline{\lambda}_{k+1}^i = \sum_{j=1}^{i}\left(\prod_{\ell=i-j+1}^{i} P_\ell^T\right)(A_{i-j+1}^T\lambda_{k+j+1} + f_{k+j}^{i-j+1}) \ .$$

Now

$$P_i^T \underline{\lambda}_{k+1}^i = P_i^T [A_i^T \lambda_{k+2} + f_{k+1}^i + \sum_{j=1}^{i-1} \left(\prod_{\ell=i-j}^{i-1} P_\ell^T \right) (A_{i-j}^T \lambda_{j+k+2} + f_{k+j+1}^{i-j})] =$$

$$= P_i^T A_i^T \lambda_{k+2} + P_i^T f_{k+1}^i + \sum_{j=1}^{i-1} \left(\prod_{\ell=i-j}^{i} P_\ell^T \right) (A_{i-j}^T \lambda_{k+j+2} + f_{k+j+1}^{i-j})$$

$$= P_i^T (A_i^T \lambda_{k+2} + f_{k+1}^i) + \sum_{j=2}^{i} \left(\prod_{\ell=i-j+1}^{i} P_\ell^T \right) \left(A_{i-j+1}^T \lambda_{j+k+1} + f_{k+j}^{i-j+1} \right)$$

$$= \sum_{j=1}^{i} \left(\prod_{\ell=i+1-j}^{i} P_\ell^T \right) \left(A_{i+1-j}^T \lambda_{j+k+1} + f_{k+j}^{i+1-j} \right) ,$$

which had to be shown.

With the convention $\prod_{\ell=i}^{i-1} P_\ell \triangleq I$, we can write (A3) for k=0 as

$$\underline{\lambda}_0^i = \sum_{j=0}^{i-1} \left(\prod_{\ell=i-j}^{i-1} P_\ell^T \right) (A_{i-j}^T \lambda_{j+1} + f_j^{i-j}) .$$

Now we can use (4.7) and (4.10) to obtain

$$\underline{\lambda}_0^i = \sum_{j=0}^{i-1} \left(\prod_{\ell=i-j}^{i-1} P_\ell^T \right) \left\{ A_{i-j}^T (-K^{-1} \sum_{k=1}^{\gamma+1} \bar{A}_k x_{j-\gamma+k}) + C_{i-j}^T R^{-1} (z_j - \sum_{k=1}^{\gamma} C_k \tilde{P}_k x_{j-\gamma+k}) \right\}.$$

Using (4.13) and (4.14) this gives

$$\prod_{\ell=i}^{\gamma} P_\ell^T \underline{\lambda}_0^i = \sum_{j=0}^{i-1} \sum_{k=1}^{\gamma+1} - D_{i-j,k} x_{j-\gamma+k} + \sum_{j=0}^{i-1} \bar{C}_{i-j}^T R^{-1} z_j .$$

In order to collect terms in x_ℓ we have to change indices. Use (A1) and the convention

$$\sum_{\ell=\gamma+1}^{\gamma} a_{\ell i} = 0$$

to arrive at ($j_1 = 1$, $j_2 = \gamma+1$, $i_1 = 0$, $i_2 = i-1$)

$$\prod_{\ell=i}^{\gamma} P_\ell^T \underline{\lambda}_0^i = \sum_{\ell=1}^{i} \sum_{k=1}^{\ell} - D_{i-\ell+k,k} x_{\ell-\gamma} +$$

$$+ \sum_{\ell=i+1}^{\gamma} \sum_{k=\ell+1-i}^{\ell} - D_{i-\ell+k,k} x_{\ell-\gamma}$$

(A4)
$$+ \sum_{\ell=\gamma+1}^{\gamma+i} \sum_{k=\ell+1-i}^{\gamma+1} - D_{i-\ell+k,k} x_{\ell-\gamma}$$

$$+ \sum_{j=0}^{i-1} \tilde{P}_{i-j}^T C_{i-j}^T R^{-1} z_j \; .$$

On the other hand, from the initial condition we know that

$$\underline{x}_0 = Q_0 \underline{\lambda}_0 + \mu_0$$

or using (4.4)

$$\underline{\lambda}_0 = Q_0^{-1} \begin{bmatrix} \tilde{P}_1 x_{-\gamma+1} \\ \vdots \\ \tilde{P}_\gamma x_0 \end{bmatrix} + Q_0^{-1} \mu_0$$

(Note: $\mu_0 = 0$ for the smoothing problem.)

Let $$S \triangleq \mathrm{diag}(\tilde{P}_1^T \ldots \tilde{P}_\gamma^T) Q_0^{-1} \mathrm{diag}(\tilde{P}_1 \ldots \tilde{P}_\gamma),$$

where
$$\hat{S} \triangleq [\mathrm{diag}(\tilde{P}_1^T \ldots \tilde{P}_\gamma^T) Q_0^{-1}]$$

These definitions allow us to write out equations for $x_{-\gamma+1} \ldots x_0$, which in turn yield $\underline{x}_0$ uniquely. Then we can combine this with (A4) to obtain a system of tne form

(A5)
$$G^0 \tilde{x}^i = G^1 \hat{x}^i + \hat{z}^i$$

where

$$\tilde{x}^i = \begin{bmatrix} x_{-\gamma+1} \\ \vdots \\ x_0 \end{bmatrix}, \; \hat{x}^i = \begin{bmatrix} x_1 \\ \vdots \\ x_\gamma \end{bmatrix}$$

(A6) $$G^0_{i,\ell} = \begin{cases} \sum_{k=1}^{\ell} D_{i-\ell+k,k} + S_{i,\ell} & \text{if } 1 \le \ell \le i \\ \sum_{k=\ell+1-i}^{\ell} D_{i-\ell+k,k} + S_{i,\ell} & \text{if } \gamma \ge \ell > i \end{cases}$$

(A7) $$G^1_{i,\ell} = \begin{cases} -\sum_{k=\ell+1-i}^{1} D_{i-\ell+k,k+\gamma} & \text{for } \ell \le i \\ 0 & \text{for } \ell > i \end{cases}$$

and the elements of $\hat{z}^i$ are given by

(A8) $$\hat{z}^i_k = \sum_{j=0}^{k-1} \bar{C}^T_{k-j} R^{-1} z_j + [\hat{S}\mu_0]_k$$

If we define

$$\Gamma_{i,\ell} = \begin{cases} \sum_{k=1}^{\ell} D_{i-\ell+k,k} & \text{if } 1 \le \ell \le i \\ \sum_{k=\ell+1-i}^{\ell} D_{i-\ell+k,k} & \text{if } \gamma \ge \ell > i \end{cases}$$

then (A6) becomes

(A9) $$G^0_{i,\ell} = \Gamma_{i,\ell} + S_{i,\ell}$$

and

$$\Gamma_{i+1,\ell+1} = \Gamma_{i,\ell} + D_{i+1,\ell+1}$$

Similarly we recognize from (4.16) and (4.17) that

$$G^1_{i,\ell} = -\tilde{A}_{\ell+\gamma-i} \qquad \text{for } \ell \le i$$

A.3 Terminal Condition

Before writing out the terminal equations, some preliminaries are needed.

Lemma:

$$\text{(A10)} \qquad \underline{\lambda}_{N-i} = \sum_{j=0}^{i} (A^T)^{i-j} f_{N-j} + (A^T)^{i+1} \underline{\lambda}_{N+1}$$

Proof: The case $i=0$ follows from (4.2) when $k=N$. Suppose the result holds for i. Then (4.2) and the induction hypothesis result in

$$\begin{aligned}
\underline{\lambda}_{N-(i+1)} &= A^T \underline{\lambda}_{N-i} + f_{N-i-1} \\
&= \sum_{j=0}^{i} (A^T)^{i+1-j} f_{N-j} + (A^T)^{i+2} \underline{\lambda}_{N+1} + f_{N-i-1} \\
&= \sum_{j=0}^{i+1} (A^T)^{i+1-j} f_{N-j} + (A^T)^{i+2} \underline{\lambda}_{N+1}
\end{aligned}$$

Let

$$\text{(A11)} \qquad A = \begin{bmatrix} 0 & P_1 & \ddots & \mathbf{0} \\ & \mathbf{0} & \ddots & P_{\gamma-1} \\ A_1 & \cdots & \cdots & A_\gamma \end{bmatrix} \text{ and } \overline{B} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \mathbf{0} & \vdots \\ 0 & \cdots & K \end{bmatrix}$$

Now we can derive equations for $\underline{x}_N$ in terms of $\underline{x}_{N-\ell}$. From (4.1) we get

$$\text{(A12)} \qquad \begin{aligned}
\underline{x}_{N-i} &= A\underline{x}_{N-i-1} + \overline{B}\underline{\lambda}_{N-i} \\
&\overset{(A10)}{=} A\underline{x}_{N-i-1} + \overline{B}\Big(\sum_{j=0}^{i} (A^T)^{i-j} f_{N-j} + (A^T)^{i+1} \underline{\lambda}_{N+1}\Big) \\
&\overset{(4.9)}{=} A\underline{x}_{N-i-1} + \overline{B} \sum_{j=0}^{i} (A^T)^{i-j} C^T R^{-1} (z_{N-j} - C\underline{x}_{N-j}) + \overline{B}(A^T)^{i+1} \underline{\lambda}_{N+1}
\end{aligned}$$

For convenience define

$$\text{(A13)} \qquad V_i = [\overline{B}(A^T)^i]_m = K[A^T]^i_m$$

and

(A14) $$W_i = V_i C^T R^{-1} \ ,$$

where the subscript m indicates the block of the last m rows.

Note that V_i is a matrix of order mxn which can be computed recursively. Taking advantage of the structure of A and $\overline{B}$ no more than n•m operations are necessary for each V_i.

Thus using (A13), (A14) and (4.6), equation (A12) gives the relation for the last m components x_{N-i} of $\underline{x}_{N-i}$ as

(A15) $$x_{N-i} = \sum_{\ell=1}^{\gamma} \overline{A}_\ell x_{N-i-1-\gamma+\ell} + \sum_{j=0}^{i} W_{i-j} z_{N-j} - \sum_{j=0}^{i} W_{i-j} \sum_{\ell=1}^{\gamma} \overline{C}_\ell x_{N-j-\gamma+\ell} + V_{i+1} \underline{\lambda}_{N+1} \ .$$

Before collecting terms, let k = -j-γ, then

(A16) $$\begin{aligned} x_{N-i} = & \sum_{\ell=1}^{\gamma} \overline{A}_\ell x_{N-i-1-\gamma+\ell} + \sum_{j=0}^{i} W_{i-j} z_{N-j} - \\ & - \sum_{k=-i-\gamma}^{-\gamma} \sum_{\ell=1}^{\gamma} W_{i+k+\gamma} \overline{C}_\ell x_{N+k+\ell} + V_{i+1} \underline{\lambda}_{N+1} \ . \end{aligned}$$

Now apply the index transformation (A1) with $j_1 = 1$, $j_2 = \gamma$, $i_1 = -i-\gamma$, $i_2 = -\gamma$. This yields an expression of the form ($k = j-\ell$)

(A17) $$\sum_k \sum_\ell a_{k,\ell} = \sum_{j=1-i-\gamma}^{-\gamma} \sum_{\ell=1}^{i+j+\gamma} a_{j-\ell,\ell} + \sum_{j=1-\gamma}^{-i} \sum_{\ell=j+\gamma}^{j+i+\gamma} a_{j-\ell,\ell} + \sum_{j=-i+1}^{0} \sum_{\ell=j+\gamma}^{\gamma} a_{j-\ell,\ell}$$

Also, since $\overline{A}_{\gamma+1} = -I_m$ we have

(A18) $$-x_{N-i} + \sum_{\ell=1}^{\gamma} \overline{A}_\ell x_{N-i-1-\gamma+\ell} = \sum_{\ell=1}^{\gamma+1} \overline{A}_\ell x_{N-i-1-\gamma+\ell} \ .$$

If we let $j = \ell-i-1$ this becomes

(A19) $$\sum_{\ell=1}^{\gamma+1} \overline{A}_\ell x_{N-i-\gamma+\ell-1} = \sum_{j=-i}^{\gamma-i} \overline{A}_{j+i+1} x_{N-\gamma+j} \ .$$

Now we can use (A17) and (A19) to rewrite (A16):

$$0 = \sum_{j=-i}^{\gamma-i} \bar{A}_{j+i+1} x_{N-\gamma+j} + \sum_{j=0}^{i} W_{i-j} z_{N-j} -$$

$$- \left[\sum_{j=1-i}^{0} \sum_{\ell=1}^{j+i} W_{i+j-\ell} \bar{C}_{\ell} x_{N-\gamma+j} + \right.$$

(A20)

$$+ \sum_{j=1}^{\gamma-i} \sum_{\ell=j}^{j+i} W_{i+j-\ell} \bar{C}_{\ell} x_{N-\gamma+j} +$$

$$\left. + \sum_{j=\gamma-i+1}^{\gamma} \sum_{\ell=j}^{\gamma} W_{i+j-\ell} \bar{C}_{\ell} x_{N-\gamma+j} \right] + V_{i+1} \underline{\lambda}_{N+1}$$

This holds for $0 \leq i \leq \gamma-1$. After collecting terms involving $x_{N-\gamma+j}$ for $j \geq 1$ on the left hand side, we obtain a system of equations of the form

$$T^0 \tilde{x}^t = T^1 \hat{x}^t + \hat{z}^t \tag{A21}$$

where

$$\tilde{x}^t = \begin{bmatrix} x_{N-\gamma+1} \\ \vdots \\ x_N \end{bmatrix}, \quad \hat{x}^t = \begin{bmatrix} x_{N-2\gamma+1} \\ \vdots \\ x_{N-\gamma} \end{bmatrix}, \quad \hat{z}^t = \begin{bmatrix} \hat{z}^t_1 \\ \vdots \\ \hat{z}^t \end{bmatrix} \text{ and}$$

$$\hat{z}^t_{i+1} = \sum_{j=0}^{i} W_{i-j} z_{N-j} + V_{i+1} \underline{\lambda}_{N+1}, \quad i = 0 \ldots \gamma-1 \tag{A22}$$

and the matrix T^0 is given by

$$T^0_{i,j} = \begin{cases} -\bar{A}_{i+j} + \displaystyle\sum_{\ell=j}^{j+i-1} W_{i-1+j-\ell} \bar{C}_{\ell}, & \text{if } 1 \leq j \leq \gamma-i+1 \\ \displaystyle\sum_{\ell=j}^{\gamma} W_{i-1+j-\ell} \bar{C}_{\ell}, & \text{if } \gamma-i+2 \leq j \leq \gamma \end{cases} \tag{A23}$$

This can be defined recursively, too. Let

(A24)
$$F_{i,j} = \begin{cases} \sum_{\ell=j}^{i+j-1} W_{i+j-\ell-1}\overline{C}_\ell \, , & 1 \le j \le \gamma-i+1 \\ \sum_{\ell=j}^{\gamma} W_{i-1+j-\ell}\overline{C}_\ell \, , & \gamma-i+2 \le j \le \gamma \end{cases}$$

Then for $1 \le j \le \gamma-i+1$

$$F_{i+1,j-1} = \sum_{\ell=j-1}^{i+j-1} W_{i+j-\ell-1}\overline{C}_\ell = W_i\overline{C}_{j-1} + F_{i,j}$$

Similarly for $j > \gamma-i+1$:

$$F_{i+1,j-1} = \sum_{\ell=j-1}^{\gamma} W_{i-1+j-\ell}\overline{C}_\ell = W_i\overline{C}_{j-1} + F_{i,j} \, .$$

Consequently F satisfies the recursion

(A25)
$$F_{i+1,j-1} = F_{i,j} + W_i\overline{C}_{j-1}$$

in both cases. Thus, after $F_{1,j}$ and $F_{i,\gamma}$ have been computed, the recursion is initialized, and

(A26)
$$T^0_{i,j} = \begin{cases} -\overline{A}_{i+j} + F_{i,j} & \text{if } 1 \le j \le \gamma-i+1 \\ F_{i,j} & \text{otherwise} \end{cases}$$

To find T^1 change the index on the right hand side from j to $k-\gamma$. Then we get

$$\sum_{k=\gamma-i}^{\gamma} \overline{A}_{i+1-\gamma+k}x_{N-2\gamma+k} - \sum_{k=1-i+\gamma}^{\gamma} \sum_{\ell=1}^{k-\gamma+i} W_{i+k-\gamma-\ell}\overline{C}_\ell x_{N-2\gamma+k} \, , \; 0 \le i \le \gamma-1$$

From this we see that for $1 \le i \le \gamma$

(A27)
$$T^1_{i,k} = \begin{cases} 0 & \text{if } k < \gamma-i+1 \\ \overline{A}_1 & \text{if } k = \gamma-i+1 \\ \overline{A}_{i-\gamma+k} - \sum_{\ell=1}^{k-\gamma+i-1} W_{i-1+k-\gamma-\ell}\overline{C}_\ell & \text{if } k > \gamma-i+1 \end{cases}$$

For $k > \gamma-i+1$ define

$$\overline{T}_{i,k} = \sum_{\ell=1}^{k-\gamma+i-1} W_{i-1+k-\gamma-\ell}\overline{C}_{\ell}$$

Then clearly

(A28) $$\overline{T}_{i,k} = \overline{T}_{i-1,k+1} ,$$

and we only need to compute $\overline{T}_{i,\gamma}$ for $i \geq 2$, i.e.,

(A29) $$\overline{T}_{i,\gamma} = \sum_{\ell=1}^{i-1} W_{i-1-\ell}\overline{C}_{\ell} \quad .$$

APPENDIX B

INVERSION OF BLOCK CIRCULANT MATRICES

A matrix of the form

$$
\text{(B1)} \qquad H_c = \begin{bmatrix} H_0 & H_1 & \cdots & H_{N-1} \\ H_{N-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & H_1 \\ H_1 & \cdots & H_{N-1} & H_0 \end{bmatrix}
$$

where H_i itself is a matrix of dimension dxd, is called a block circulant matrix with dimension N and base dimension d.

Consider the first column of blocks H_0, H_{N-1}, ... H_1, and form sequences of length N by taking the elements in the position (i,j) of each matrix, i.e., $H_0(i,j)$, $H_{N-1}(i,j)$, ... $H_1(i,j)$).

Then the inverse of H_c can be computed efficiently in the following manner:

Step 1: Take the DFT of the sequences $H_0(i,j)$, $H_{N-1}(i,j)$ ... $H_1(i,j)$ for each (i,j) and denote those transforms by $\hat{H}_0(i,j)$, $\hat{H}_{N-1}(i,j)$, ... $\hat{H}_1(i,j)$.

Step 2: Compute the inverses

$$\hat{B}_k \hat{=} \hat{H}_k^{-1} \text{ , for } k = 0, \ldots N-1$$

Step 3: Take the inverse DFT of the sequences

$$\hat{B}_0(i,j), \hat{B}_{N-1}(i,j), \ldots \hat{B}_1(i,j)$$

and denote the result by

$$B_0(i,j), B_{N-1}(i,j), \ldots B_1(i,j)$$

which gives the first column of blocks of H_c^{-1}.

Step 4: Circulate the blocks B_k to obtain the matrix $B = H_c^{-1}$ given by

$$B = \begin{bmatrix} B_0 & B_1 & \cdots & B_{N-1} \\ B_{N-1} & \ddots & \ddots & B_1 \\ & \ddots & \ddots & \\ B_1 & & B_{N-1} & B_0 \end{bmatrix}$$

Proof: Since H_c is a NxN block circulant matrix with base d, H_c can be expressed as a sum of Kronecker product of matrices as

(B3) $$H_c = \sum_{k=0}^{N-1} C_k \otimes H_k$$

(B4) $$C_0 = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 1 & & & 0 \end{bmatrix}, \ldots, \quad C_{N-1} = \begin{bmatrix} 0 & & 0 & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$$

where each C_k is NxN and each H_k is dxd. Define

(B5) $$\hat{H}_c = F\, H_c\, F^* = (F \otimes I_d)\left(\sum_{k=0}^{N-1} C_k \otimes H_k\right)(F^* \otimes I_d)$$

where F is the NxN unitary discrete Fourier transform matrix. Then by properties of the cross project of matrices, we get

(B6) $$\hat{H}_c = \sum_{k=0}^{N-1} (F \otimes I_d)(C_k \otimes H_k)(F^* \otimes I_d)$$

(B7) $$= \sum_{k=0}^{N-1} (F\, C_k \otimes H_k)(F^* \otimes I_d)$$

(B8) $$= \sum_{k=0}^{N-1} (F\, C_k\, F^*) \otimes H_k$$

Since C_k is a circulant matrix with 1's and zeros, the product $F\ C_1\ F^*$ is simply a diagonal matrix, D_k, i.e.,

$$\text{(B9)} \qquad D_k = F\ C_k\ F^* = \begin{bmatrix} d_k(0) & & & 0 \\ & d_k(1) & & \\ & & \ddots & \\ 0 & & & d_k(N-1) \end{bmatrix}$$

where $d_k(\ell) = e^{-j\frac{2\pi}{N}k\ell}$ for $k,\ell = 0,1, \ldots N-1$ (B10)

Hence $\hat{H}_c = \sum_{k=0}^{N-1} D_k \otimes H_k$ (B11)

$\hat{H}_c$ is a block diagonal matrix with the diagonal blocks as $\hat{H}_\ell$, i.e.,

$$\hat{H}_\ell = \sum_{k=0}^{N-1} d_k(\ell) H_k \ .$$

Substituting (B10) results in

$$\hat{H}_\ell(i,j) = \sum_{k=0}^{N-1} e^{-j\frac{2\pi}{N}k\ell} H_k(i,j)$$

Since $\hat{H}_c$ is a block diagonal matrix, the inverse of $\hat{H}_c$ is simply the inverse of each block matrix on the main diagonal.

In order to get B, the inverse of H_c, the inverse DFT is required. Note that

$$(\hat{H}_c)^{-1} = (F\ H_c\ F^*)^{-1}$$

$$\hat{H}_c^{-1} = (F^*)^{-1} H_c^{-1} (F)^{-1}$$

$$H_c^{-1} = F^* \hat{H}_c^{-1}\ F = F^{-1} \hat{H}_c^{-1} (F^{-1})^*$$

The inverse of a block circulant matrix is also a block circulant matrix. This can be shown as follows:

$\hat{B}$ can be expressed as $$\hat{B} = \sum_{k=0}^{N-1} E_k \otimes \hat{B}_k$$

where
$$E_0 = \begin{bmatrix} 1 & 0 \cdots 0 \\ 0 & \\ \vdots & \underline{0} \\ 0 & \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 & 0 & \cdots \\ 0 & 1 & \\ \vdots & & 0 \\ \cdot & & \end{bmatrix}, \quad \ldots, \quad E_{N-1} = \begin{bmatrix} 0 & & \\ & \ddots & 0 \\ & 0 & 1 \end{bmatrix}$$

Then $$B = H_c^{-1} = (F^* \otimes I_d)\left(\sum_{k=0}^{N-1} E_k \otimes \hat{B}_k\right)(F \otimes I_d)$$

$$= \sum_{k=0}^{N-1} (F^* E_k F) \otimes \hat{B}_k$$

It is found that $F^* E_k F$ is a circulant matrix. Therefore $(F^* E_k F) \otimes \hat{B}_k$ is a block circulant matrix. Because summation of block circulant matrices gives a block circulant; B is a block circulant matrix.

REFERENCES

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Trans. ASME, Ser. D., J. Basic Engng. 82, 35-45, 1960.

[2] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," Trans. ASME, Ser. D., J. Basic Engrg. 83, 95-108, 1961.

[3] T. Kailath, "A View of Three Decades of Linear Filtering Theory," IEEE Trans. Inform. Theory IT-20, Mar. 1974.

[4] J. S. Meditch, "A Survey of Data Smoothing for Linear and Nonlinear Dynamic Systems," Automatica 9, 151-162, 1973.

[5] D. G. Lainiotis, "Estimation: A Brief Survey," Inform. Sci. 7, 3, 191-202, 1974.

[6] Y. C. Ho and R. C. K. Lee, "A Bayesian Approach to Problems in Stochastic Estimation and Control," IEEE Trans. of Aut. Control, 9, 1964.

[7] H. E. Rauch, F. Tung and C. T. Strieble, "Maximum Likelihood Estimation of Linear Dynamic Systems," AIAA, 3, 1965.

[8] J. S. Meditch, "Orthogonal Projection and Discrete Optimal Linear Smoothing," SIAM J. Control, 5, 1967.

[9] T. Kailath, "An Innovations Approach to Least Square Estimation, Pt. I: Linear Filtering in Additive White Noise," IEEE Trans. Aut. Control, AC-13, Dec. 1968.

[10] D. G. Lainiotis, "Partioned Estimation Algorithm, II: Linear Estimation," Information Sciences, 7, 1974.

[11] D. C. Fraser, "A New Technique for the Optimal Smoothing of Data," MIT Instrumentation Lab. T474, Cambridge, Mass., 1967.

[12] J. E. Potter and R. G. Stearn, "Statistical Filtering of Space Navigation Measurements," Proc. 1963 AIAA Guidance and Control Conf., 1963.

[13] P. Dyer and S. McReynolds, "Extension of Square Root Filtering to Include Process Noise," J. Optimiz. Theory Appl., Vol. 3, 1969.

[14] S. F. Schmidt, "Computational Techniques in Kalman Filtering," NATO-AGARD-139, Feb. 1970.

[15] P. G. Kaminski and A. E. Bryson, "Discrete Square Root Smoothing," Proc. AIAA Guidance and Control Conf. 1972.

[16] G. J. Bierman, "Sequential Square Root Filtering and Smoothing of Discrete Linear Systems," Automatica, 10, Mar. 1974.

[17] T. Kailath, "Some New Algorithms for Recursive Estimation in Constant Linear Systems," IEEE Trans. Inform. Theory, IT-19, Nov. 1973.

[18] M. Morf, et. al., "Some New Algorithm for Recursive Estimation in Constant Linear, Discrete Time System," IEEE Trans. on Aut. Control, AC-19, Aug. 1974.

[19] A. G. Carlton, "Linear Estimation in Stochastic Processes," Bumblebee Series Rept. No. 311, Johns Hopkins University, Silver Spring, MD, 1962.

[20] H. E. Rauch, "Solutions to the Linear Smoothing Problem," IEEE Trans. Aut. Control, AC-8, 1963.

[21] A. E. Bryson and M. Frazer, "Smoothing for Linear and Nonlinear Dynamic Systems," TDK 63-119, Aero. Sys. Div., Wright Patterson Air Force Base, Ohio, 1963.

[22] D. Q. Mayne, "A Solution to the Smoothing Problem for Linear Dynamic System," Automatica 4, 1966.

[23] B. D. O. Anderson and S. Chiratrattananon, "New Linear Smoothing Formulas," IEEE Trans. Aut. Control, AC-17, Feb. 1972.

[24] W. F. Trench, "Inversion of Toeplitz Band Matrices, Math. of Computation, Vol. 28, Oct. 1974.

[25] N. Levinson, "The Wiener RMS Error Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, 261-278, Jan. 1947.

[26] N. E. Nahl, Estimation Theory and Applications, John Wiley and Son, Inc., 1969.

[27] A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, 1970.

[28] N. K. Sinha and P. Rozsa, "Some Canonical Forms for Linear Multivariable Systems," Int. J. Control, Vol. 23, 1976.

[29] A. K. Jain, "Fast Inversion of Banded Toeplitz Matrices by Circular Decompositions," IEEE Trans. on ASSP, ASSP-26, 2, 121-126, April 1978.

[30] K. W. Au, "Linear Estimation Via Fast Fourier Transform," M.S. Thesis, SUNY, Buffalo, Sept. 1979.

[31] B. D. O. Anderson and Moore, Optimal Filtering, Prentice Hall, Inc., 1979.

DATE

ILME

—8